
Dobusch ▪ Forsterleitner ▪ Hiesmair (Hg.): Freiheit vor Ort



Leonhard Dobusch
Christian Forsterleitner
Manuela Hiesmair
(Hg.)

Freiheit vor Ort

Handbuch kommunale Netzpolitik

Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grunde sind die in dem vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor(en), Herausgeber, Übersetzer und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht, auch nicht für die Verletzung von Patentrechten, die daraus resultieren können. Ebenso wenig übernehmen Autor(en) und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. werden ohne Gewährleistung der freien Verwendbarkeit benutzt und können auch ohne besondere Kennzeichnung eingetragene Marken oder Warenzeichen sein und als solche den gesetzlichen Bestimmungen unterliegen.

Bibliografische Information Der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

2011, Open Source Press, München

Bearbeitete Neuauflage des Bandes „Freie Netze. Freies Wissen.“,

Echo media verlag ges.m.b.h., Wien 2007.

Sämtliche Texte dieses Buches stehen unter der Lizenz „Creative Commons, Namensnennung – Weitergabe unter gleichen Bedingungen 2.0“

<http://creativecommons.org/licences/by-sa/2.0/at/legalcode>

Gesamtlektorat: Dr. Markus Wirtz

Satz: Open Source Press (L^AT_EX)

Umschlaggestaltung: Olga Saborov, Open Source Press

Gesamtherstellung: Kösel, Krugzell

ISBN (E-Book, PDF) 978-3-941841-40-6

<http://www.opensourcepress.de>

4 Kapitel

Freie Software im öffentlichen Sektor: Chancen und Aufgaben auf lokaler Ebene

Leonhard Dobusch und Jakob Huber

„Ubuntu“ ist ein Wort der afrikanischen Sprache Zulu und steht dort für „Menschlichkeit“ und „Gemeinsinn“, aber auch für den Glauben an ein „universelles Band des Teilens, das alles Menschliche verbindet“. In Europa wurde dieses Wort aber erst in den letzten Jahren bekannt als Name einer populären Version des Computerbetriebssystems *Linux*. Von seinem größten Konkurrenten Windows unterscheidet sich Ubuntu weniger in seiner Benutzerführung oder in seinen Funktionen, sondern durch zwei andere Eigenschaften: Es ist kostenlos erhältlich. Und jedeR, der/die möchte, darf es verändern, erweitern und an Freunde und Bekannte weitergeben. Diese Unterschiede zu Windows machen auch deutlich, warum der süd-

afrikanische Ubuntu-Gründer Marc Shuttleworth diesen speziellen Namen für die Software ausgewählt hat: Er unterstreicht, dass Ubuntu nur durch die gemeinschaftliche Zusammenarbeit vieler Menschen möglich ist, die zwar über die ganze Welt verstreut, aber durch das Internet miteinander verbunden sind.

Die Ubuntu Community ist nur eine von unzähligen weltweit, die sich der gemeinschaftlichen Erstellung von Freier bzw. Open Source Software verschrieben haben.¹ Nahezu jedes kommerzielle Computerprogramm hat mittlerweile freie Open-Source-Pendants, die nicht nur kostenlos erhältlich, sondern teilweise sogar funktional überlegen sind. Der freie Internet-Browser *Mozilla Firefox* löst auf immer mehr PCs Microsofts Internet Explorer ab. Statt Briefe in Word und Tabellen in Excel zu erstellen, nutzen immer mehr Menschen die Programme Writer oder Calc des freien Programms *OpenOffice* (Tabelle 4.1 zeigt eine Auflistung der wichtigsten Softwareprogramme).

Tabelle 4.1:
Tabelle 1: Verbreitete Software und ihre kostenlosen Pendants mit freiem Quellcode

Closed Source Software	Free/Libre/Open Source Software
Microsoft Windows, Apple MacOS	GNU/Linux (z.B. Ubuntu, Suse, Red-Hat, ...)
MS Internet Explorer, Opera, Safari	Mozilla Firefox, Chromium, Konqueror
MS Outlook (Express), Lotus Notes	Mozilla Thunderbird
MS Office (Word/Excel/Powerpoint/Access)	OpenOffice.org/LibreOffice (Writer/Calc/Impress/Base)
Adobe Photoshop	GIMP
Adobe Illustrator, Quark Xpress	Scribus, Inkscape
MS Media Player	VLC Media Player
eDonkey, µTorrent, Bittorrent	eMule, LimeWire, Vuze
GIF-Format, MP3-Format	PNG-Format, OGG-Format

Dieses Phänomen, nämlich dass Menschen Ergebnisse ihrer Arbeit frei und für jedeN zugänglich machen, hat (nicht nur) ÖkonomInnen verblüfft und ratlos gemacht: Warum beteiligen sich so viele Menschen an der Entwicklung von freier Software? Wieso funktioniert dieses Entwicklungsmodell? Ist Open Source Software vielleicht sogar eine Gefahr für die Wirtschaft, eine Art Cyber-Kommunismus im Internet? Darf und kann mit freier Software Geld verdient werden?

Mit Fragen und Vorwürfen dieser Art schlägt sich der Pionier der Open-Source-Bewegung, Richard Stallman, seit der Gründung der Free Software

¹ Freie Software ist nach der Definition der Free Software Foundation (<http://www.fsf.org>) immer auch Open Source Software. Umgekehrt muss das nicht gelten und hängt wesentlich davon ab, welcher Software-Lizenz das Programm unterliegt.

Foundation (FSF) regelmäßig herum (siehe auch Interview in diesem Buch). Zumindest auf die Frage nach den Gründen für Einzelne, sich an der gemeinschaftlichen Erzeugung von freier Software zu beteiligen, hat er eine ausführliche Antwort parat: Die Gründe reichen vom reinen Spaß am Programmieren über Nächstenliebe und Dankbarkeit bis hin zu politischem Idealismus oder doch auch Geld.² Letzteres steht für Stallman keineswegs im Widerspruch zum Freiheitsideal. Wenn jemand mit frei zugänglicher Software Geld verdienen kann, dann soll er oder sie das auch dürfen, solange damit nicht die gemeinschaftlichen Rechte und die gemeinsame Weiterentwicklung eingeschränkt werden.

Wesentlichste Voraussetzung ist der Zugang zum Quelltext (Source Code) eines Computerprogramms, der als Sammlung für (fachkundige) Menschen lesbarer Anweisungen an den Computer Auskunft gibt, warum und wie Software funktioniert. Zu Beginn des Computerzeitalters war in diesem Sinne jede Software „Open Source.“

4.1 Eine kurze Geschichte freier Software

So beginnt auch der Gründungspräsident der *Free Software Foundation Europe*, Georg Greve, Vorträge zum Thema Freie Software meist mit folgendem Satz: „Am Anfang war alle Software frei.“ Zeitlich lag dieser Anfang in den 1960er Jahren. Computer gab es nur an wenigen Forschungsinstituten, wie zum Beispiel den „Labs“ des *Massachusetts Institute of Technology* (MIT), und an den Räume füllenden Maschinen arbeiteten große Teams. Für die ForscherInnen war es selbstverständlich, jede Idee, jeden Verbesserungsvorschlag, jede Programmzeile mit KollegInnen zu teilen und gegenseitig zu überprüfen („Peer Review“). Software war kein eigenes Produkt, sondern wurde eher als „Zugabe“ zur Hardware verstanden – praktisch jedes Computerprogramm war damals „Open Source“.

In dieser Zeit etablierte sich auch die „HackerInnen-Kultur“, wobei die Bezeichnung „HackerInnen“ damals keinen negativen, sondern eher einen anerkennenden Beigeschmack hatte. Dem 2001 in Bielefeld verstorbenen Computer-Aktivisten der ersten Stunde Wau Holland wird dabei folgende Erklärung zugeschrieben:

Wenn man die Kaffeemaschine benutzt, weil der Herd nicht geht, um Wasser heiß zu machen, das dazu verwendet wird, die Fertigmischung für Kartoffelbrei zuzubereiten, dann ist man ein Hacker.

² Vgl. Möller, E. (2005): Die heimliche Medienrevolution – Wie Weblogs, Wikis und freie Software die Welt verändern, S. 63f.

Auch das damals leistungsfähigste Betriebssystem Unix war maßgeblich von den HackerInnen einiger amerikanischer Universitäten entwickelt worden. Der US-Telekommunikationskonzern AT&T besaß zwar die Rechte daran, dessen kartellrechtliche Probleme ermöglichten es aber den EntwicklerInnen, auch Kopien des für Menschen lesbaren Unix-Quellcodes weiterzugeben. Die folgenden Jahre waren von großen Fortschritten bei der Leistungsfähigkeit, Stabilität, Portabilität auf verschiedene Systeme und vor allem die Fähigkeit sich an Netzwerke anzuschließen, geprägt.

Als aber mit dem Aufkommen der PCs ein immer größerer Markt für Software entstand, begannen Softwarefirmen, ihre Programme unter Lizenz zu stellen und nur mehr den für Menschen unlesbaren Maschinen-Code („Binaries“) weiterzugeben. Auch AT&T startete 1983 den ausschließlichen Vertrieb seiner Unix-Version als proprietäre Software. Eine an der Universität Berkeley entwickelte, freie Unix-Version namens BSD³ sollte gleichzeitig durch Klagen vom wachsenden Markt verdrängt werden. Freie Software schien als Fußnote der Softwaregeschichte zu enden, die immer mehr von HerstellerInnen proprietärer Software wie Apple oder Microsoft geschrieben wurde.

Manche HackerInnen wollten sich aber nicht mit der Situation abfinden, dass andere mit den Ergebnissen ihrer Arbeit Profite einfuhren und sie selbst nicht einmal mehr Zugriff auf den Quellcode hatten. Mehr noch, sie wollten auch verhindern, dass sich eine Geschichte wie die rund um Unix noch einmal wiederholen könnte. Einer dieser Hacker war Richard Stallman vom MIT. Er begann noch im selben Jahr mit dem GNU-Projekt (GNU steht als rekursives Akronym für „GNU's Not Unix“) zur Entwicklung eines freien Unix-Klons, der der Community *frei* zur Verfügung stehen sollte. Um das Projekt dauerhaft zu schützen, entwickelte Stallman das *Copyleft*-Prinzip, das Urheberrechte gerade dafür einsetzt, die freie Verwendbarkeit des Werkes zu garantieren. Zwei Jahre später gründet er als Trägerin des GNU-Projekts und zur Förderung freier Software die *Free Software Foundation*. Gemeinsam mit dem Rechtsprofessor Eben Moglen fasste Stallman schließlich 1989 verschiedene Copyleft-Lizenzen unter der „GNU General Public License“ (GPL) zusammen. Damit waren alle wichtigen Grundsteine freier Software gelegt, zum endgültigen Durchbruch fehlte aber noch etwas: das Internet.

Bevor es im Internet einfach und schnell möglich war, immer wieder immer neue Versionen des Quellcodes einer breiten und über die ganze Welt verstreuten EntwicklerInnengemeinde zukommen zu lassen, mussten die PionierInnen freier Software Disketten mit dem Quellcode per Post verschicken. Entsprechend langsam verlief die Weiterentwicklung in der Zeit vor dem Internet. Dass der Aufstieg des heute größten und bekanntesten Stücks freier Software, Linux, mit dem des Internets zusammenfällt, ist somit auch alles andere als ein Zufall.

³ Berkeley Software Distribution

Weil er gerne Unix-Software auf (relativ) billiger PC-Hardware verwenden wollte, bastelte der finnische Student Linus Torvalds an einem eigenen Betriebssystemkern namens „Freax“ und stellt erste Ergebnisse auf einem Internet-Server zur Verfügung. Da dem Verantwortlichen des Servers der Name Freax nicht gefiel, änderte er den Namen des Verzeichnisses auf „Linux“ – Linus' Unix. Zusammen mit Software aus dem GNU-Projekt, dem genau so ein Kern noch gefehlt hatte, gab es damit plötzlich ein völlig freies Betriebssystem und jede Menge Software für PCs. Nach einem Seminar bei Stallman in Helsinki stellte Torvalds auch Linux unter die GPL-Lizenz des GNU-Projekts und veröffentlichte 1994 schließlich Linux in der Version 1.0. Zum offiziellen Logo von Linux wurde ein Pinguin auserkoren, dessen Name Tux für „Torvalds' Unix“ steht und gleichzeitig ein Wortspiel mit der englischen Bezeichnung für Frack („Tuxedo“ oder einfach „Tux“) ist.

4.2 Free vs. Open Source Software

Während in den 1990er Jahren mit dem Internet auch die Community von NutzerInnen und EntwicklerInnen freier Software exponentiell wuchs, wurde ihr im Schatten von Microsofts Aufstieg zum profitabelsten Konzern der Welt zumindest in der Geschäftswelt nur geringe Aufmerksamkeit zuteil. Die doppelte Bedeutung von „free“ als „frei“ und „gratis“ im Englischen hatte daran sicher ihren Anteil.

Nach der Niederlage im „Browserkrieg“ gegen Microsoft und der Übernahme durch AOL gab die Firma Netscape 1998 den Quellcode des vormaligen Marktführers unter den Internetbrowsern, *Netscape Communicator*, unter dem Projektnamen „Mozilla“ frei.⁴ Eine Gruppe von HackerInnen wollte diese Situation für eine Verankerung freier Software im Businessbereich nutzen. Einer von ihnen, Eric Raymond, formuliert es rückblickend so:

Wir begriffen, dass Netscapes Ankündigung ein kostbares Zeitfenster geöffnet hatte, in dem es uns endlich gelingen könnte, die Unternehmenswelt dazu zu bringen, sich anzuhören, was wir ihr über die Überlegenheit eines offenen Entwicklungsmodells beizubringen hatten. Wir erkannten, dass es an der Zeit war, die Konfrontationshaltung abzulegen, die in der Vergangenheit mit der „freien Software“ in Verbindung gebracht wurde, und die Idee ausschließlich mit den pragmatischen, wirtschaftlichen Argumenten zu verkaufen, die auch Netscape dazu motiviert hatte.

⁴ Der zu komplizierte Code wurde von der Community vereinfacht und aufgeteilt und vier Jahre später unter den Namen „Firefox“ (Browser), „Thunderbird“ (Mail-Programm) und „Sunbird“ (Kalendersoftware) veröffentlicht und weiterentwickelt.

Noch im selben Jahr folgte das Gründungstreffen der *Open Source Initiative* (OSI). Der Vorschlag, statt von „Freier“ nur noch von „Open Source Software“ zu sprechen, geht auf Christine Peterson,⁵ einer der (bislang) wenigen führenden Frauen in der Open-Source-Bewegung, zurück. Die Gründungsversammlung sah damit ein Problem gelöst: Die Zweideutigkeit von „frei“ (free) im Sinne von Freibier und/oder Freiheit. Außerdem war damit eine Abgrenzung zu dem als (zu) „ideologisch“ betrachteten Ansatz der Free Software Foundation (FSF) verbunden. Diese sah aber wiederum die *Freiheit* als Kern ihrer Identität an und befürchtete mittelfristig eine Schwächung freier Software. Die Diskussionen über die Werte und Ziele Freier Software spaltete die Community, und bis heute bestehen ein *Free-Software-* und ein *Open-Source-Lager*.

Wer dabei an die Spaltungen sozialer Bewegungen in den 1960er Jahren denkt, hat nur teilweise Recht. Zwar stimmen beide Seiten darin überein, dass sie unterschiedliche Prinzipien haben.

Open Source ist eine Entwicklungsmethode, Free Software ist eine soziale Bewegung. [...] Für die Open-Source-Bewegung ist nicht-freie Software eine suboptimale Lösung. Für die Free-Software-Bewegung ist sie ein soziales Problem und Free Software die Lösung.⁶

So lautet der gängige Vergleich, den die Free Software Foundation verwendet. Dabei ist man sich „in den praktischen Ableitungen und Handlungsempfehlungen mehr oder weniger einig.“ Diese Einschätzung spiegelt sich auch im „Alltag“ der Entwicklung freier Software wider, wo die Unterschiede keine große Rolle spielen und auch die wenigsten sich genau einem Lager zuordnen würden.

Open Source Software

Open Source Initiative (OSI, <http://www.opensource.org>) veröffentlicht Kriterien als Grundlage für ein „Open-Source“ Gütesiegel. Quelltexte müssen offen, veränderbar und verbreitbar sein. OSI präsentiert sich bewusst unpolitisch. Bekannteste Lizenz: Mozilla Public License.

Free Software (Freie Software)

Bezeichnet Software, die in der Regel dem Copyleft unterliegt (dadurch automatisch auch Open Source, aber nicht umgekehrt). Free Software Foundation fördert nicht nur Verbreitung, sondern auch Aufklärung. Bekannteste Lizenz: GNU General Public License (GPL)

⁵ Die Chemikerin versucht in ihrem Brotberuf für das Foresight Institut Medien- und Politikverantwortliche über Nanotechnologie und ihre Langzeitfolgen aufzuklären. Ihr Ziel ist ein Technologieeinsatz zum Wohle der Umwelt und der Menschen.

⁶ GNU Operating System (2007): Why „Free Software“ is better than „Open Source“, online: <http://www.gnu.org/philosophy/free-software-for-freedom.html>

Freeware

Kostenlose Programme, deren Quellcode nicht unbedingt frei zugänglich ist.

Public Domain

Software, an der alle UrheberInnenrechte abgegeben worden sind. Damit auch nicht z. B. durch Copyleft „geschützt.“

Ein Grund dafür ist, dass nach der Definition der Open Source Initiative (OSI) Freie Software immer auch Open Source Software ist. Die drei Grundpfeiler dieser Definition sind nämlich weniger streng als die Definition Freier Software der Free Software Foundation: Erstens müssen die Quelltexte vorliegen, zweitens muss das Programm beliebig kopiert, verbreitet und genutzt werden dürfen und drittens muss das Programm verändert verbreitet werden dürfen. Ohne *Copyleft* fehlt den Open-Source-Lizenzen aber die Auflage, veränderte Programme der Öffentlichkeit zur Verfügung zu stellen.

Ironischerweise ist in jüngster Zeit verstärkt auch die Vieldeutigkeit von „Open Source“ ein Problem: Quellcode, der einsehbar ist, aber nicht verändert werden darf, ist zwar „quelloffen“, aber eben nicht „Open Source“ im Sinne der Definition. Ein Open-Source-Gütesiegel konnte sich wohl auch deshalb nie durchsetzen, da selbst überwiegend proprietäre Software in den Regalen als „Open Source“ verkauft wurde. Bruce Perens, Autor der *Open Source Definitions* und OSI-Mitgründer, wechselte nach einem Jahr dennoch zufrieden die Lager. Die OSI habe ihre Aufgabe, der Nicht-HackerInnen-Welt die freie Software nahezubringen, erfüllt:

Und jetzt ist es Zeit für die zweite Phase: Jetzt, wo alle Welt zusieht, ist es für uns an der Zeit, Sie über Freie Software aufzuklären. Beachten Sie, ich sagte Freie Software und nicht etwa Open Source.

4.3 Viele Augen sehen mehr

So umstritten die „richtige“ Lizenz und die „richtige“ Bezeichnung unter ProgrammiererInnen freier Software ist, so überzeugt sind sie in der Regel von den Vorzügen ihres Entwicklungsmodells. Auch wenn sich freie und proprietäre Software auf den ersten Blick nur durch die Softwarelizenz unterscheiden, sind die Konsequenzen dieses Unterschieds für ihre Herstellung („Softwareentwicklung“) enorm. Während proprietäre Software ähnlich herkömmlicher Industrieprodukte in Firmen unter größter Geheimhaltung des Quellcodes gefertigt, getestet und schließlich ausgeliefert wird, dreht sich dieser Prozess bei freier Software quasi um und ähnelt am Ende mehr und mehr wissenschaftlichen Methoden. Aber der Reihe nach.

Am Anfang eines freien Softwareprojekts steht meist jemand, der ein Problem hat und seine (rudimentäre) Lösung anderen zugänglich macht. Die Veröffentlichung steht somit am Anfang und nicht am Ende des Softwareprojekts. Leute mit ähnlichen oder gleichen Problemen müssen in der Folge nicht mehr bei Null beginnen und das sprichwörtliche Rad neu erfinden, sondern können auf der bestehenden Vorarbeit aufbauen. Indem andere die Software unter anderen Bedingungen (andere Hardware, andere Betriebssystemversion etc.) anwenden, treten oftmals Fehler zu Tage, die dem/der ursprünglichen EntwicklerIn verborgen geblieben wären. Solche „Bugs“ genannten Softwarefehler können nun im Internet einfach zurückgemeldet oder – dank des offenliegenden Quellcodes – gleich selbst korrigiert werden. Zumindest potentiell ist damit die Dauer und Vielfalt der Fehlertests bei freier Software jener bei proprietärer Software weit überlegen. In seinem richtungsweisenden Aufsatz „Die Kathedrale und der Basar“⁷ prägte Eric Ramyond entsprechend den Satz „Given enough eyes, all bugs are shallow“ – wenn nur genug EntwicklerInnen hinschauen, wird jeder Fehler gefunden.

Aber nicht nur Bugs, auch die vorgeschlagene Lösung selbst steht nach ihrer Veröffentlichung in der Kritik der jeweiligen *Community*. Ist sie voller Fehler – also völlig „verbugt“ – kann es durchaus sein, dass jemand einen anderen Vorschlag zur Bewältigung des gleichen Problems liefert. Wer hilfreiches Feedback auf „seinen“ Code bekommen möchte, ist außerdem angehalten, sauber und übersichtlich zu programmieren. Alles Umstände, die dazu beitragen, die Qualität des Quellcodes und damit des Computerprogramms zu heben. Und alles Umstände, die dem Ideal wissenschaftlicher Forschung relativ nahe kommen. Auch dort sollen Ergebnisse so bald wie möglich veröffentlicht und der Kritik zugänglich gemacht werden. Auch dort sollen gegenseitige Anregungen und Kritik die Ergebnisse besser, robuster machen. Gerade dort ist man sich der Bedeutung des Satzes „Standing on the Shoulders of Giants“⁸ bewusst, nämlich dass Fortschritte und Weiterentwicklung auf der Vorarbeit zahlreicher anderer davor aufbauen. So braucht es nicht mehr zu verwundern, dass Open-Source-ProgrammiererInnen genauso wie ForscherInnen Communities bilden: In beiden Gruppen stehen die Mitglieder zwar im Wettbewerb zueinander, erreichen aber nur durch Kooperation ihr gemeinsames Ziel.

Gleichzeitig ist die Entwicklung freier Software noch einen Tick demokratischer als der Wissenschaftsbetrieb: Wer sich dem „Mainstream“ nicht anschließen und ein Softwareprojekt lieber in eine andere Richtung weiterentwickeln möchte, der kann das auch tun. Voraussetzung dafür ist nur, genügend MitstreiterInnen zu finden, um das Projekt am Leben zu halten. In

⁷ Vgl. Ramyond, E. (1999): Die Kathedrale und der Basar, online: <http://gnuwin.epfl.ch/articles/de/Kathedrale/> [26.08.2006].

⁸ Mehr über den Spruch, der auch Googles Wissenschaftssuchmaschine „Google Scholar“ zitiert: http://en.wikipedia.org/wiki/Stand_on_the_shoulders_of_giants [16.08.2006].

der Geschichte des bereits erwähnten freien Betriebssystems BSD-Unix gab es gleich mehrere derartiger, „Fork“ genannter, Spaltungen der Community: Während *NetBSD* die Unterstützung vielfältigster Hardwareplattformen ins Zentrum rückte, konzentrierten sich die *OpenBSD*-EntwicklerInnen vor allem auf Sicherheitsaspekte und ihre KollegInnen von *FreeBSD* auf größtmögliche Freiheit des Quellcodes. Die Projekte befruchten und ergänzen sich dabei – dank des offenen Quellcodes – gegenseitig.

Sämtliche Vorzüge des kollektiven Entwicklungsmodells freier Software sind in den Grundregeln freier Softwarelizenzen bereits angelegt: Indem sie zur Offenlegung von Weiterentwicklungen an freier Software verpflichten, ermöglichen sie erst eine umfassende Qualitätskontrolle in der Community und liefern gleichzeitig den Anreiz, eigenen Code auch offenzulegen.

4.4 Was nichts kostet ist nichts wert?

Viele der Eigentümlichkeiten freier Softwareentwicklung münden unmittelbar in Vorteile auf Seiten der AnwenderInnen. Der Wegfall der Lizenzkosten ist dabei – vor allem bei größeren Unternehmen oder Verwaltungen – noch der geringste Vorzug. Viel wichtiger ist beispielsweise der Sicherheitsaspekt: Jeder Programmierfehler ist eine potentielle Sicherheitslücke. Selbst wenn freie Software nicht besser programmiert ist als proprietäre,⁹ so werden durch den offenen Quellcode Fehler in der Regel schneller gefunden und korrigiert. Dieses Konzept der Sicherheit durch Transparenz steht diametral dem proprietären Ansatz der Sicherheit durch Verdunkelung („security through obscurity“) entgegen und wird von öffentlichen Stadtverwaltungen wie Wien oder München als ein Grund für den verstärkten Einsatz freier Software angeführt.

Ebenfalls direkt mit dem Zugang zum Quellcode ist die große Flexibilität freier Software verbunden, „was nicht passt, wird passend gemacht“ könnte das Motto lauten. Dabei wird aber gleichzeitig deutlich, dass freie Software nur selten kostenlos ist. Für die Anpassung von Software an individuelle Bedürfnisse wollen natürlich ProgrammiererInnen beschäftigt und bezahlt werden. Im Gegensatz zu herkömmlicher Software gibt es dann aber zahlreiche AnbieterInnen derartiger Dienstleistungen, und man ist nicht auf den guten Willen und den Kostenvoranschlag des Herstellers angewiesen.

Herstellerunabhängigkeit spielt nicht nur bei der Anpassung an individuelle Bedürfnisse eine große Rolle. Die Möglichkeit, Lizenzgebühren zu sparen und gleichzeitig die jeweils lokale Softwareindustrie zu fördern, hat freie Software vor allem in Entwicklungs- und Schwellenländern populär

⁹ Studien weisen darauf hin, dass die Fehlerhäufigkeit beispielsweise bei Linux geringer ist als bei Windows. Vgl. Der Standard Online (2005): Studie: Linux hat deutlich weniger Fehler, online: <http://derstandard.at/?url=/?id=1891556> [18.01.2011].

gemacht. Der forcierte Einsatz von Linux in Schulen und öffentlicher Verwaltung in Brasilien nach dem Wahlsieg des linken Präsidentschaftskandidaten Luiz Inácio Lula da Silva entsprang derartigen Überlegungen. Dieser musste sich aber am Weltsozialforum 2005 im brasilianischen Porto Allegre dennoch die Kritik gefallen lassen, dass Brasilien immer noch mehr Geld für Softwarelizenzen als für den Kampf gegen den Hunger ausbebe.¹⁰ Mittlerweile ist Open Source Software in der Verwaltung, im Bildungswesen und in der Privatwirtschaft im Vormarsch. Brasilien ist heute neben Indien das Zentrum der Open-Source-Entwicklung außerhalb von Europa und Nordamerika.

Wie fundamental die (auch: mittelbaren) Auswirkungen eines Umstiegs auf Open Source Software sein können, durfte die Stadt München nach ihrer Entscheidung zum Wechsel von Windows NT auf Linux am Behörden-Desktop im Jahr 2003 erfahren: Auf einmal konnte (und musste) die Vergabe der Softwareumgebung wieder formal ausgeschrieben werden, und es fand ein Wettbewerb diverser AnbieterInnen statt. Gleichzeitig war man so sehr an die Dominanz eines Herstellers gewohnt, dass hunderte größere und kleinere Fachanwendungen erst mühsam plattformunabhängig gemacht werden mussten. Dieser Aufwand wurde aber in dem Wissen in Kauf genommen, sich nicht neuerlich in die Sackgasse einer Abhängigkeit von einem Hersteller zu begeben: Durch die Wahl der völlig freien Linux-Distribution *Debian* ist sichergestellt, dass die Stadtverwaltung auch in Zukunft aus einer Reihe von AnbieterInnen und DienstleisterInnen wählen wird können.

Hinzu kommt noch, dass auch andere öffentliche Behörden unmittelbar von der Münchner Umstellung auf freie Software profitieren: Softwarelösungen, die im Zuge der Umstellung für München neu programmiert wurden, stehen Dank Open-Source-Lizenz automatisch auch anderen Stadtverwaltungen kostenlos zur Verfügung. So ist aus der Münchner OpenOffice-Erweiterung „Wollmux“ – ein Wortspiel aus „eierlegende Wollmilchsau“ und „Linux“ – inzwischen bereits ein eigenes Open-Source-Projekt samt Webseite unter <http://www.wollmux.org> geworden.¹¹

4.5 Geld verdienen mit „Gratis“ Software

Der Vorteil, bei der Verwendung freier Software aus einer Vielfalt an AnbieterInnen auswählen zu können, ist auch ein Hinweis auf zahlreiche Varianten mit vermeintlich kostenlosen Programmen Geld zu verdienen. Das ist wenig überraschend, da auch HerstellerInnen proprietärer Software oft

¹⁰ Vgl. Süddeutschen Zeitung Online, online: <http://www.sueddeutsche.de/kultur/artikel/75/47028/index.html>.

¹¹ Details über die Linux Migrationen in München und Wien finden sich bei: Dobusch, L. (2008): Windows versus Linux.

mit Service- und Dienstleistungsverträgen viel mehr als mit den Lizenzen selbst verdienen. In Unternehmen oder der öffentlichen Verwaltungen ist die Betreuung der Software ohnehin eine laufende Aufgabe.

Spätestens aber seit auch die „Big Player“ der High-Tech-Industrie, wie IBM, Intel oder der Datenbankspezialist Oracle, verstärkt auf freie Software setzen, zweifeln nur noch die wenigsten an deren wirtschaftlichem Potential. Die von Oracle übernommene Firma Sun Microsystems leistete sich sogar einen offiziellen Open-Source-Evangelisten („Chief Open Source Officer“) namens Simon Phipps. Zu Besuch in Deutschland bei einer Konferenz mit dem Titel „Open Source meets Business“ lieferte der auch gleich eine ganze Liste mit Möglichkeiten, wie sich mit freier Software auch (gut) bezahlte Arbeitsplätze schaffen lassen. Im Folgenden findet sich Phipps Aufzählung von Open-Source-Software-Geschäftsmodellen, die von dualen Lizenzen über Betreuung und Beratung bis hin zur „versteckten“ Nutzung freier Software in Hardware-Produkten reicht.

Duale Lizenz

Neben der freien Software mit entsprechender Lizenz wird weiterhin ein proprietäres Produkt unter proprietärer Lizenz gepflegt. Verbesserungen am freien Produkt werden laufend in das proprietäre integriert, das zusätzlich um spezifische Fähigkeiten erweitert wird (z. B. Rechtschreibprüfung in Oracles OpenOffice-Klon StarOffice).

Abonnement

Kombination von betreuter freier Software mit einem Supportangebot: Die AbonnentInnen erhalten laufend und automatisch immer die neuesten Updates (z. B. RedHat oder Novell-Suse Linux).

Betreuung/Hosting

Die Software wird von den AnbieterInnen betrieben, angeboten wird der Service (z. B. kommerzielle Wikis/Datenbanken auf Basis freier Wiki/Datenbank-Software)

Beratung/Consulting

Beschränkung auf Beratung beim Einsatz freier Softwareprodukte

Embedded-Bereich

Einsatz von freier Software gemeinsam mit einem verkaufbaren Produkt, insbesondere Hardware. Die Palette reicht dabei von der Waschmaschine über den Handheld (PDA) bis hin zur Playstation III von Sony. Besonders bemerkenswert ist hier der Smartphone-Markt, wo freie Betriebssysteme wie Googles Android oder Nokia und Intels „MeeGo“ immer mehr Marktanteile gewinnen.

„Stewardship“

Entwicklung und Kontrolle von (offenen) Standards, finanziert von den NutznießerInnen dieser Standards.

Aber auch wenn es vor allem ums Geldverdienen geht, bleiben immer noch gar nicht so kleine Unterschiede zwischen freier und proprietärer Software bestehen. Die größte Ironie ist dabei, dass die unmittelbar auf Kooperation angewiesene freie Software zu stärkerem Wettbewerb in allen Bereichen der Wertschöpfungskette von Software führt. Denn das Programmieren des Quellcodes ist nur eine von mehreren geldwerten Leistungen rund um Software.

Freie Betriebssysteme wie Linux sind dafür ein gutes Beispiel: Prinzipiell sind alle Bestandteile auch der umfangreichsten Version frei und kostenlos im Internet als Download erhältlich. Die meisten haben aber nicht den Überblick oder die Zeit, um sich selbst ihre Wunschsoftware zusammenzusuchen. Firmen wie RedHat oder Novell-Suse nehmen sich dieser Arbeit an, bündeln die Softwareteile aus verschiedensten Quellen und stellen sie zu praktischen Komplettpaketen zusammen, die sie dann zum Beispiel gemeinsam mit Handbüchern verkaufen. An der Freiheit der Software ändert sich dadurch natürlich nichts.

Wer Lust hat, kann jede einzelne Komponente rund um freie Software, vom Quelltext über die Handbücher bis hin zu regelmäßigen Updates von verschiedenen Firmen beziehen und manche Aufgaben davon selbst übernehmen. Letzteres ist eine Strategie, die die Wiener Stadtverwaltung rund um ihr „Wienux“-Projekt gewählt hat: Wienux ist eine an die spezifischen Wiener Bedürfnisse angepasste Version von Linux und wurde fast zur Gänze in der hauseigenen Magistratsabteilung (MA) 14 entwickelt. Dass es auch komplett anders geht, beweisen die Münchner Kollegen, die für die Entwicklung des Behördendesktop „Limux“ nach breiter Ausschreibung zwei mittelständische Firmen der Region beauftragten.

Wertschöpfungsbereiche im Softwarebusiness

- Quelltext/Sourcecode
- Ausführbare Software/Binaries
- Dokumentation
- Fehlerbehebung
- Schulung/Ausbildung
- Garantie
- Haftung (z.B. wegen Patentverletzungen)
- Support (z.B. regelmäßige Sicherheitsupdates)

Bei *proprietärer Software* muss in der Regel der größte Teil der Wertschöpfungskette von einem Hersteller/einer Herstellerin bezogen werden.

Bei *Open Source Software* können im Unterschied zu proprietärer Software sämtliche wertschöpfenden Bereiche von verschiedenen HerstellerInnen bezogen werden. Gleichzeitig sind in allen Bereichen die Markteintrittsbarrieren relativ gering, da kein hoher Kapitalaufwand notwendig ist.

In beiden Fällen, sowohl dem der Wiener als auch dem der Münchner Stadtverwaltung, gilt aber, dass der verstärkte Einsatz freier Software eine diskrete Form lokaler Wirtschaftsförderung sein kann.

4.6 Hürden am Weg zur freien Software-Welt

Trotz der Initiativen in München oder Wien ist der Einsatz freier Software im großen Stil immer noch alles andere als eine Selbstverständlichkeit. Neben Wissensdefiziten und Vorurteilen gegenüber dem früheren „Hacker“-Betriebssystem Linux gibt es dafür eine Reihe von Gründen. Zentral ist sicherlich, dass Softwaremärkte Netzwerkmärkte sind. Je mehr Menschen ein Programm verwenden, desto besser in der Regel auch für den/die individuelleN AnwenderIn. Einfacherer Dateiaustausch, größere Auswahl an Zusatzprogrammen und mehr ausgebildete Fachkräfte sprechen meist für das Produkt mit dem größten Marktanteil. Neue (Open Source) Software hat außerdem mit dem Problem zu kämpfen, dass potentielle BenutzerInnen ihr bereits erworbenes Know-how bei einem Wechsel nicht zur Gänze auf die Alternative übertragen können.

Diese strukturellen Schwierigkeiten sind aber noch längst nicht alles, was einer freieren Softwarelandschaft entgegensteht. Die größte Gefahr droht von juristischer Seite: Wie jede Freiheit ist auch die von Software auf rechtlichen Schutz angewiesen. Das bislang für Software zuständige Urheberrecht leistete hier in Form des *Copyleft* auch gute Dienste. Doch Bestrebungen und Lobbying vor allem finanzstarker Konzerne zur Patentierbarkeit von Quellcode könnten der Dynamik freier Software schnell ein Ende bereiten. Das Nachprogrammieren bestimmter Funktionen wäre dann nämlich nur noch gegen Lizenzgebühren möglich. Die nahtlose Integration patentierter Algorithmen wie des populären Musikformats MP3 und des Bildformats GIF in freie Software ist schon heute aus genau diesem Grund schwierig. Europa muss vor Softwarepatenten bewahrt werden, fordert daher auch der Freie-Software-Pionier Richard Stallman bereits seit Jahren.¹²

Eine ähnliche Bedrohung sind geschlossene, proprietäre Standards und Formate, die aber nicht nur freie Software, sondern auch Institutionen wie das Internet bedrohen. Ein wesentlicher Grund für den Erfolg des Inter-

¹² Vgl. Stallman, R. (1999): Europa muß vor Softwarepatenten bewahrt werden, online: <http://www.heise.de/tp/deutsch/special/wos/6437/1.html> [18.07.2006].

nets sind seine völlig offenen und freien Kommunikationsstandards,¹³ die es den verschiedensten Computern mit den unterschiedlichsten Betriebssystemen ermöglichen, miteinander zu kommunizieren. In gleichem Maße ist die proprietäre Natur des .doc- bzw. .docx-Formats von Microsofts Word eine Hürde für alle (Open Source) Konkurrenten, die mühsam versuchen müssen, solche Dateien zu importieren.

Denn logischerweise räumen etablierte Platzhirsche wie Microsoft oder Adobe ihre komfortable Position nicht freiwillig. So finanziert Microsoft eine Unmenge an Studien, um die Unwirtschaftlichkeit von freier Software zu „beweisen“, und bedient sich einer als „FUD“ (Fear, Uncertainty, and Doubt) bekannten Taktik gezielter Desinformation und Verunsicherung.

4.7 Von destruktivem zu kooperativem Wettbewerb

Die verstärkte Nutzung freier Software in München und Wien war auch begleitet von großem medialem Trommelwirbel, der nur am Rande mit technischen Details der Umstellungen zu erklären ist. Vielmehr wurde die politische Bedeutung des Einsatzes von freier Software zum Thema gemacht. Während die einen in der Entscheidung für GNU/Linux einen ideologisch motivierten Akt linker Stadtverwaltungen sahen, ging anderen das Engagement der beiden Städte für freie Software noch lange nicht weit genug. Letzteres war sicherlich auch Folge der städtischen Kommunikationspolitik sowohl in München als auch in Wien: In beiden Fällen bemühten sich die PolitikerInnen, jede politische und ideologische Komponente ihrer Entscheidung weit von sich zu weisen, und betonten ihre rein technisch-wirtschaftlichen Beweggründe.

Abgesehen von den Untiefen vergaberechtlicher Regelungen stellt sich natürlich die Frage, ob die Förderung freier Software eine öffentliche Aufgabe ist oder sein sollte. Geht es hier nur um verschiedene Technologien, gar nur um verschiedene Produkte? Oder *ist* die Verwendung freier Software auch eine politische Frage, spricht man doch von Software als „öffentlichem Gut“? Für die Free Software Foundation und ihren Präsidenten Richard Stallman ist die Antwort hier klar: Freie Software ist ein Menschenrecht, das politisch gesichert werden muss. Je wichtiger Software für das Leben in modernen Internet-Gesellschaften wird, desto wichtiger ist auch der freie und gleiche Zugang zum Quelltext.

Diesen politischen Willen vorausgesetzt bleibt die Frage, welche Möglichkeiten die öffentliche Hand zur Förderung freier Software hat. Eine zentrale Rolle spielt hier, welche Software in Schulen und anderen Bildungsein-

¹³ Das gilt sowohl für das Datenübertragungsprotokoll TCP/IP als auch für die Auszeichnungssprache HTML.

richtungen eingesetzt wird. So ist es nicht Großzügigkeit oder Nächstenliebe, sondern purer Geschäftssinn, dass Microsoft seine Software an Schulen und Universitäten extrem billig oder sogar zum Nulltarif lizenziert. Die Herstellung der dafür notwendigen Kopien kostet nichts, der Return on Investment in Form späterer, auf Microsoft konditionierter AnwenderInnen ist enorm. Denn auch Unternehmen legen sich natürlich jene Programme zu, mit denen ihre MitarbeiterInnen schon in der Schule gelernt haben umzugehen.

So hat beispielsweise in Österreich das Bildungsministerium 2003 stolz einen Generalnutzungsvertrag mit Microsoft präsentiert und diesen 2008 verlängert. Die rund 50.000 PCs im Besitz der Bundesschulen mitsamt den dazugehörigen Servern und die Privat-PCs der LehrerInnen werden mit der kompletten Microsoft-Produktpalette (Windows, Office etc.) ausgestattet. Microsoft erhält dafür vom Bildungsministerium jährlich mindestens 2,5 Millionen Euro und jahrgangswise SchulabsolventInnen, die in der Regel nur Microsoft-Produkte kennen gelernt haben. Die Schule hat aber nicht die Möglichkeit, auch alle SchülerInnen mit der Software für den Heimcomputer auszustatten. Wenn Linux oder Open Source Software zum Einsatz kommt, dann nur auf Initiative einzelner LehrerInnen. Das gilt für alle mittleren und höheren Schulen, die Situation für die Pflichtschulen unterscheidet sich davon kaum.

So kommt auch eine Auftragsstudie des österreichischen Wirtschaftsministeriums zu dem Schluss, dass Open-Source-ExpertInnen in der Regel einen akademischen IT-Hintergrund oder die Kenntnisse zu einem großen Teil autodidaktisch erworben haben. Die ExpertInnen für proprietäre Produkte stammen hingegen meist von einer Fachhochschule oder HTL.

Wie es anders gehen könnte, zeigt das Beispiel Schleswig-Holstein, wo das Landesbildungsamt mit *kmLinux* eine für den Bildungsbereich adaptierte Linux-Distribution entwickelte. Die erste Auflage in der Höhe von 5.000 CDs war binnen Wochen vergriffen. Schulen in Schleswig-Holstein werden kostenlos ausgestattet. Viele der Anwendungen, die im Laufe des Projekts entwickelt und gesammelt wurden, sind dabei web-basiert und somit völlig plattformunabhängig – sie funktionieren sowohl mit Linux als auch mit Windows. Einige deutsche (vorwiegend technische) Schulen nutzen den Einsatz von Open Source Software auch zur Lösung der schulischen IT-Probleme: SchülerInnen arbeiten direkt an der Software und verbessern zum Beispiel die hauseigene Verwaltungssoftware und werden dafür prämiert. Auch die eingangs erwähnte Linux-Distribution *Ubuntu* hat mit *Edu-buntu* einen eigenen Ableger speziell für Ausbildungszwecke gegründet.

Bei all diesen Initiativen fällt auf, dass sie sich nicht in die Dichotomie Privat vs. Staat zwingen lassen. Wenn die öffentliche Hand auf freie Software setzt und damit zu Software als öffentlichem Gut beiträgt, hat das nichts mit „Verstaatlichung“ zu tun. Im Gegenteil, gerade freie Software ist auf individuelle Initiative und Beiträge angewiesen. Gleichzeitig kann niemand

vom Gebrauch ausgeschlossen werden – bei freier Software handelt es sich per Definition um ein Gemeingut. Verschiedene freie Softwareprogramme stehen in starkem Wettbewerb um die beste Lösung. Die schaffen aber wiederum diejenigen, die am meisten EntwicklerInnen und AnwenderInnen zur *Kooperation* bewegen können. Der Wettbewerb bei der Entwicklung freier Software ist in diesem Sinne ein kooperativer und produktiver. An (und in) der öffentlichen Hand liegt es, diesen kooperativen Wettbewerb zu ermöglichen und zu fördern.

„Die Bewegung für freie Software ist eine Bewegung für Menschenrechte und für soziale Veränderung.“

Interview mit Richard M. Stallman

Der Pionier freier Software Richard Stallman startete in den 80er Jahren am Massachusetts Institute of Technology (MIT) das GNU-Projekt (ein wesentlicher Bestandteil jeder Linux-Distribution), entwarf mit der GPL (General Public License) die wichtigste Copyleft-Lizenz für freie Software und gründete 1985 der Free Software Foundation, deren Präsident er heute noch ist.



So kurz wie möglich: Was ist die politische und philosophische Basis von Freier Software?

Freie Software bedeutet, die Freiheit der Nutzer zu respektieren. Es gibt vier grundsätzliche Freiheiten, die Freie Software definieren: Freiheit 0: Die Freiheit, die Software so zu verwenden, wie man will. Freiheit 1: Die Freiheit, den Quelltext zu lesen und das Programm so zu verändern, wie man will. Freiheit 2: Die Freiheit, anderen Kopien des Programms zu geben. Das entspricht der Freiheit, deinem Nächsten zu helfen. Freiheit 3: Die Freiheit, modifizierte Versionen zu vertreiben. Das ist die Freiheit, deiner Gemeinschaft zu helfen.

Das sind Menschenrechte, die jedeR SoftwarenutzerIn haben sollte. Es ist falsch, irgendjemandem diese Rechte zu verwehren. Proprietäre Software – also nicht-freie Software – verschafft den Entwicklern eine Machtposition

über die Nutzer und führt unter ihnen zu Uneinigkeit und Hilflosigkeit. Das ist falsch, es ist ein soziales Problem. Unser Ziel ist, dieses Problem zu lösen.

Glauben Sie, man kann von einer „Bewegung für Freie Software“ sprechen? Würden Sie zustimmen, dass es sich dabei um eine soziale Bewegung handelt?

Definitiv. Die Bewegung für freie Software ist eine Bewegung für Menschenrechte und für soziale Veränderung. Und als solche habe ich sie von Beginn an gesehen.

Wie definieren Sie die Beziehung von Freier Software zu ähnlichen Bereichen wie „Freiem Wissen“ oder „Freier Kultur“? Was sind die Ähnlichkeiten? Was sind die Unterschiede?

In den 90er Jahren kam ich zum Schluss, dass alle geschriebenen Werke, die einen praktischen Zweck für die Gesellschaft erfüllen, in gleichem Sinne frei sein sollten. Werke für praktische Zwecke inkludiert Software, Kochrezepte, Lehrbücher und Nachschlagewerke. Wenn man Texte für praktische Aufgaben in seinem Leben verwendet, muss man die Kontrolle darüber haben.

Dieses Argument gilt jedoch nicht für Kunstwerke oder Meinungsäußerungen. Ich glaube nicht, dass diese auch frei sein müssen. Ich glaube aber sehr wohl, dass jedeR das Recht haben muss, exakte Kopien dieser Werke für nichtkommerzielle Zwecke weiterzugeben.

Ihrer Meinung nach, was kann eine Stadt wie Linz mit ca. 200.000 EinwohnerInnen tun, um sowohl das Konzept bzw. Idee von als auch Freie Software an sich zu unterstützen?

Sie kann in der Stadtverwaltung zu Freier Software wechseln. Aber viel wichtiger ist es, in den Schulen Freie Software zu verwenden und zu unterrichten. Ich weiß aber nicht genau, wie groß der Einfluss der Stadtverwaltung auf die Schulen in Österreich ist.

Wie können Städte und Stadtverwaltungen im allgemeinen zum Universum Freier Software beitragen?

Gleiche Antwort wie vorhin. Außerdem können größere Städte die Entwicklung von speziellen Programmen finanzieren, die sie für ihre Verwaltungsarbeit brauchen.

Wie würden Sie die Rolle der Universitäten in der Entwicklung von mehr und besserer Freier Software einschätzen?

Alle Schulen sollten ihre SchülerInnen und Studierenden helfen, als Mitgliedern einer freien Gesellschaft zu leben, und das bedeutet auch die Verwendung Freier Software. Das gilt wiederum für Universitäten genauso wie es für Grundschulen gilt.

Im Jahr 2005 bekam die Free Software Foundation einen Anerkennungspreis bei der Prix Ars Electronica in Linz. Ist mit Freier Software eine spezielle „Kunst“ oder „Kultur“ verbunden?

Ich glaube, Software ist eine Handwerkskunst, verglichen mit den schönen Künsten. In diesem Sinne also ja. Kultur wiederum sind alle Muster von Handlungen und Gedanken, die zwischen Menschen ausgetauscht werden. Software ist somit Kultur, egal ob sie frei ist oder nicht.

Was sind die Gefahren für den weiteren Erfolg und die weitere Verbreitung Freier Software? Was sind die Herausforderungen?

Das größte Hindernis sind Gesetze in vielen Ländern, die Freie Software verbieten. In Frankreich wurde kürzlich ein Gesetz verabschiedet, das den Besitz einer Kopie von DeCSS [ein freies Programm zur Entfernung des DVD-Kopierschutzes, Anm.] zu einem Verbrechen macht. Gleichzeitig will die Weltorganisation für geistiges Eigentum (World Intellectual Property Organisation – WIPO) bei der Überarbeitung des „Broadcast-Treaty“ die Verwendung Freier Software für softwareunterstütztes Radio quasi unmöglich machen, und die Europäische Union will Radioübertragungen im Internet ebenfalls reglementieren. Warum verabschieden scheinbar demokratische Regierungen solche Gesetze? Weil die Demokratie von Mega-Konzernen untergraben wird.

Sie betonen immer den Unterschied zwischen Freier Software und Open Source Software. Warum?

Die Bewegung für Freie Software basiert auf den ethischen Grundsätzen der Freiheit und der Gemeinschaft. Als GNU/Linux und andere Freie Software in den 90er Jahren voll entwickelt war, stellte sie sich als machtvoll und verlässlich heraus, und Millionen von Menschen begannen sie zu verwenden. Aber viele von denen teilten nicht unsere Werte, viele hatten sogar nie von ihnen gehört.

Der Begriff „Open Source“ wurde 1998 kreiert mit dem Zweck von Freier Software zu sprechen, ohne unsere Ideale zu erwähnen. Die Open-Source-Kampagne zitiert praktische Werte, wie die Qualität und Robustheit, aber sie verschweigt ethische Werte wie Freiheit. Deshalb gibt es in unserer Gemeinschaft jetzt zwei Lager, das Freie-Software-Lager und das Open-Source-Lager.

Im Bereich der praktischen Aktivitäten machen beide Lager ähnliche Arbeit; Menschen mit verschiedenen Ansichten arbeiten zusammen. Das Open-Source-Lager hat weichere Kriterien als wir, weshalb manche Open-Source-Lizenzen nicht kompatibel mit Lizenzen Freier Software sind. Da diese aber nicht sehr weit verbreitet sind, sind praktisch alle Open-Source-Programme auch Freie Software.

Falls wir uns nur mit der Entwicklung Freier Software beschäftigen würden, wäre es vielleicht von Bedeutung, welche der beiden Philosophien die

Leute vertreten. Aber wenn wir eine Gesellschaft mit nachhaltiger Freiheit schaffen wollen, dann müssen wir vor allem den Wert der Freiheit betonen und auch andere lehren, ihn zu schätzen.

Was sind die Hauptgründe für die explosionsartige Verbreitung Freier Software, ihr unglaubliches Wachstum in den letzten Jahren?

Ich kann nur raten, aber vielleicht liegt es daran, dass wir jetzt bequeme graphische Benutzeroberflächen haben, so dass Freie Software genauso einfach zu lernen ist wie proprietäre Software.

Wie würden Sie einen Nutzer, der mit seinem Computer einfach nur arbeiten will und sich überhaupt nicht für technische Details interessiert, von einem Wechsel hin zu Freier Software überzeugen?

Ich verwende meine Energien nicht darauf, die Leute von einem Wechsel zu Freier Software zu überzeugen. Viele andere Leute tun das, so dass ich mich mit etwas wichtigerem beschäftigen kann. Ich versuche die Ideale der Freiheit und der Gemeinschaft den Millionen von Menschen näher zu bringen, die bereits Freie Software verwenden oder darüber nachdenken, sie zu verwenden, aus welchem Grund auch immer.

Zum Abschluss, bitte nennen Sie uns noch Ihre persönlichen Lieblingssoftwareprojekte der letzten Jahre.

Ich versuche nicht bei den aktuellsten Programmen von Freier Software auf dem Laufenden zu bleiben, da ich die meiste Zeit nur Texte schreibe, Mails, Programme und Artikel. Diese Arbeit erledige ich mit GNU Emacs. Man könnte also sagen, GNU Emacs ist mein liebstes Stück Freie Software.

Wenn ich an Freie Softwarepakete denke, dann denke ich nicht hinsichtlich meiner eigenen Softwarebedürfnisse. Ich denke daran, was die Community braucht, um der Freiheit zum Triumph zu verhelfen. Gnash, GNU Classpath, GCJ und GNOME sind in diesem Sinne einige der wichtigsten freien Programme.

„Die Stadt als Inkubator.“

Interview mit Gerald Kempinger

Gerald Kempinger ist Geschäftsführer der Informations- und Kommunikationstechnologie Gesellschaft (IKT), einem Unternehmen der Stadt Linz. Die IKT betreut die gesamte Unternehmensgruppe Stadt Linz. Seit 2010 leitet er das Projekt „Open Commons Region Linz“. Zuvor studierte er Informatik und arbeitete viele Jahre in verschiedensten Managementfunktionen für IBM in Österreich und Deutschland.



Herr Kempinger, Sie sind Mitherausgeber der Studie „Open Commons Region Linz“. Welche persönlichen Erfahrungen haben Sie mit Open Source Software (OSS)?

Wir haben in der Stadt Linz schon sehr frühzeitig OSS im Einsatz gehabt. Einerseits natürlich bei Servern im Web- und in verschiedensten IT-Managementbereichen, und zusätzlich haben wir bereits ab 2007 parallel zu den Microsoft-Produkten auch Open-Source-Produkte am Arbeitsplatz eingesetzt: Firefox als alternativer Browser, Open Office (OO) für ODF-Eingangsdokumente oder neuerdings auch OpenCMS als Content Management System sowie Lösungen mit PostgreSQL-Datenbanken. In Summe schätze ich, dass wir rund 40 Open-Source-Produkte im Einsatz haben, aber das ist heutzutage in IT-Unternehmen durchaus normal. Es ist auch relativ einfach, Open-Source-Office-Komponenten einzusetzen, wenn es keine Abhängigkeiten zu Unternehmensanwendungen, also Geschäftsprozesslösungen gibt, die bei der Stadt Linz zum Beispiel mit Produkten wie SAP abgedeckt werden.

Im Schulbereich werden die Arbeitsplätze durch die Stadt Linz bereits seit einigen Jahren standardmäßig mit OO ausgerüstet. Ansonsten wird es nur dann verwendet, wenn man es braucht, um in der Kommunikation mit BürgerInnen offene Dokumente zu bearbeiten. Eine Stadt muss, wenn sie

mit BürgerInnen kommuniziert, natürlich alle Formate empfangen und unterstützen können. Der flächendeckende Einsatz von Open Office ist aber derzeit nicht geplant. Es hat sich herausgestellt, dass man gänzlich ohne Microsoft-Produkte nicht auskommt und die Microsoft-Lizenzpolitik den Kauf einzelner Produkte ohne Office tendenziell daher sehr teuer gestaltet. Im innerbetrieblichen Ablauf stößt der Einsatz der Open-Office-Produkte an die Grenzen, etwa bei der zentralen Wartbarkeit. Wir als IKT-Dienstleister der Unternehmensgruppe Stadt Linz betreuen rund 6.000 Arbeitsplätze, bei der zentralen Verwaltung und Verteilung von Updates gibt es leider bei Open Office noch viele Defizite. Wir haben auch Richtung Oracle Open Office inkl. deren kostenpflichtigen Zusatzprodukten überlegt, aber letztendlich sind wir zum Ergebnis gekommen, dass Linz vorerst weiterhin auch die Microsoft-Office-Produkte für die Stadt Linz internen Arbeitsplätze einsetzen wird.

Welche Gründe sprechen für den Einsatz von Open Source Software?

Erstens leiden wir mit Sicherheit bei proprietären Programmen sehr oft unter der HerstellerInnenabhängigkeit und den entsprechenden Lizenz- und Supportpolitiken. Es gibt gerade für betriebliche Geschäftsprozessanwendungen heute leider wenige, aber letztendlich sehr dominante AnbieterInnen. Nicht nur Microsoft, man muss auch die Abhängigkeit von SAP und anderen Lösungsanbietern sehen – diese AnbieterInnen haben natürlich gewinnorientierte Vorgehensweisen, und wenn KundInnen diese Lösung für unternehmerische Kernprozesse im Einsatz haben, begeben sie sich in eine große Abhängigkeit. Die HerstellerInnen diktieren die Support Modelle, bestimmen die Preise und auch die funktionalen Erweiterungen. Man muss dann diesen HerstellerInnen „nachhüpfen“ – damit sind viele laufende Kosten verbunden. Man verliert Flexibilität, und letztendlich ist auch die Investitionskraft des Unternehmens durch die stets steigenden, laufenden Kosten eingeschränkt.

Zweitens haben sich der Support durch die Open Source Communities durchaus bewährt. Probleme werden rasch behoben und Verbesserungen vorgenommen – da muss man der Community ein Kompliment machen: Niemand sagt mehr, Open Source kann man nicht einsetzen, weil es keinen Support gibt. Man muss nur mit den Communities entsprechend umgehen lernen. Aber es gelingt, die HerstellerInnenabhängigkeit zu reduzieren und über geringere Wartungsaufwände die Kosten durchaus auch zu senken. Aber zu sagen, Open Source ist gratis, ist definitiv falsch.

Welche Erfahrungen haben Sie in Bezug auf die Akzeptanz durch die MitarbeiterInnen?

Die EndbenutzerInnen haben die Aufgabe, ihre Geschäftsprozesse effizient abzuwickeln. Alles, was sie dabei unterstützt und das Leben einfacher macht, ist in Ordnung und wird angenommen. Alles, was keine Vorteile bringt und einen Umlernprozess verlangt, wird von vornherein eher abge-

lehnt. Bei Office-Produkten kommt jetzt wieder die Dominanz des größten Herstellers hinzu, denn fast alle MitarbeiterInnen sind es gewohnt, schon beginnend bei ihrer Schulausbildung mit Microsoft-Produkten zu arbeiten. Was die BenutzerInnen aber mittlerweile verstärkt sehen ist, dass dort auch gehörig Geld verschwindet, speziell auch für den Einsatz im Privatbereich. Da die Funktionalität von Open Office, Gimp, TYPO3 etc. ja in der Zwischenzeit unbestritten ist, werden somit gerade für PrivatanutzerInnen sowie Klein- und Mittelbetriebe die Open-Source-Office-Produkte immer interessanter.

Einen Punkt möchte ich aber noch erwähnen: Die Open Office Community ist derzeit aufgrund von Einflussnahmen großer Hersteller (Sun-Übernahme durch Oracle) in Turbulenzen bzw. in einer Neustrukturierungsphase, und man muss abwarten, wie sich speziell die Libre Office Community und deren Produkte in den nächsten Jahren weiterentwickeln.

Wo sehen Sie Schwierigkeiten beim Umstieg zu Open Office aus Endnutzersicht?

Erstens haben gerade die jüngeren MitarbeiterInnen das Verständnis, dass man Open-Source-Produkte einsetzen soll, aber in der Regel haben sie auch Microsoft-Produkte in der Ausbildung verwendet. Zweitens bekommt man sehr selten Dateien in offenen Formaten zugesendet. Natürlich kann Open Office jede Microsoft-Office-Datei öffnen, aber komplexere Dokumente bleiben nicht 1:1 bei der Formatierung, nicht zuletzt weil Unternehmen wie Microsoft sich eben nicht wirklich an die offenen Standards halten.

Was kann dagegen unternommen werden?

Ich glaube, dass sich demnächst das Umfeld komplett verändert. Auch im Office-Bereich werden Lösungen sich rasch durchsetzen, die wir nicht mehr auf unseren PCs installieren, sondern die wir über das Internet nutzen. Wenn man an die Welle, die Google mit Google Mail und Google Docs ausgelöst hat, denkt und wie Microsoft versucht mit Produkten wie Office365 nachzuziehen oder daran, dass Oracle Open Office auch in nächster Zeit auch als Weboffice verfügbar gemacht werden soll, dann sieht man, dass hier eine ganz neue Bewegung hineinkommt. Das verändert das ganze Management, den Support und das Datei-Standard-Thema. Dann arbeiten wir mit Browser- oder Apps-basierten Lösungen, die in der „Cloud“ installiert sind, und haben die Möglichkeit von überall und mit jedem beliebigen Endgerät darauf zuzugreifen. Das wird sicher auch rasch im Businessbereich interessant. Es wird nicht mehr der klassische PC sein, auf dem man lokale Software installiert. Das Endgerät, ob PC, Tablet, Screen oder Smartphone, wird endlich benutzerfreundlich, d. h. die Technologie tritt in den Hintergrund. Derzeit sehen wir bereits den Beginn bei den Apple-iOS-, Android- und Chrome-OS-Geräten und -Lösungen. Da kommt es dann auch in den Unternehmen zu einem ganz großen Umdenken in den nächsten fünf bis zehn Jahren.

In welchen Bereichen müsste die Community daher einen Entwicklungsschwerpunkt setzen?

Es gibt noch sehr wenige Konzepte, in denen sich die Open Source Community mit Cloud-Lösungen auseinandersetzt. Echte Open Cloud Standards sind noch nicht wirklich definiert. Heute gibt es üblicherweise einen Sourcecode, der von der Community gemeinsam entwickelt wird, jedeR hat Zugriff und kann Fehler beheben und anderen wieder zur Verfügung stellen. Das basiert weitgehend auch von der Lizenzierung her auf dem Modell lokal auf PC- oder Server-installierter Software. Die Welt verändert sich aber in eine andere Richtung, es geht um Web- bzw. Cloud- Lösungen, die man nicht selber installiert, sondern als Service nutzt. Und das auch mehr und mehr aus firmeneigenen Netzen heraus. Da muss die Community sich rasch überlegen, was ist eine Open Cloud und wie bringt sie mehr und mehr Open-Source-Lösungen für die Cloud zustande.

Cloud klingt faszinierend und nebulös, aber dahinter stehen große Facilities, Rechenzentren, die von Google, Amazon, Microsoft etc. betrieben werden. Man braucht dafür die entsprechenden Kapazitäten auf der ganzen Welt verteilt, eine Community ist zur Finanzierung einer eigenen Infrastruktur normalerweise nicht in der Lage. Man wird daher immer eine Zusammenarbeit mit großen, kommerziellen PartnerInnen eingehen müssen...

...oder mit der öffentlichen Hand?

Oder mit der öffentlichen Hand, wenn es eine Österreich-Cloud oder eine EU-Cloud gäbe, in der man ebenfalls Lösungen hosten lassen kann. Zumindest als Alternative zu den oft auch mit Werbung vollgestopften Gratisangeboten der großen Infrastruktur-Provider. Ein ganz kleines und lokales Beispiel ist hier der Public Space Server, mit dem die Stadt Linz bereits seit einigen Jahren BürgerInnen oder Vereinen kostenlosen Webspace und Facilities anbietet, natürlich in einem für eher private Anforderungen definierten Ausmaß. Die großen Open Source Communities bräuchten aber sehr oft eine Allianz mit Microsoft, Amazon, Google, HP, Oracle oder sonst einem kommerziellen Anbieter, der in der Lage ist, eine weltweite und skalierbare Infrastruktur zur Verfügung zu stellen. Die Frage, ob wirklich die öffentliche Hand eine Alternative bieten soll und konkurrenzfähige Open Cloud Facilities zur Verfügung stellen kann, ist derzeit noch schwer zu beantworten.

Zurück zur Studie zur Open-Commons-Region, bei der es ja um viel mehr als „nur“ die Software geht. Was zeichnet die Studie aus, was sind die wichtigsten Ergebnisse?

Wir haben uns mit dem Thema freier Zugang zu Informationen, Materialien, Werken, Daten ganzheitlich befasst und es nicht etwa auf Software eingeschränkt. Andere Regionen oder europäische Initiativen sind sehr stark auf Open Source Software fixiert. Die entscheidende Empfehlung der Studie

ist, dass Linz eine Open Commons Region aufbauen soll, die weit über den Großraum hinaus wirkt – digitales Gemeingut kennt ja keine Stadtgrenzen, in der digitalen Welt gibt es keine unüberwindbaren Grenzen. Linz soll der Ausgangspunkt sein, um hier Neues entstehen zu lassen. Es soll der ideale Nährboden für Unternehmen oder Communities sein, die sagen: Hier gibt es perfekte Rahmenbedingungen, hier wird man unterstützt, hier gibt es Pilotprojekte und entsprechende Grundüberlegungen, so dass man neue Dinge umsetzen kann. Kurz: Hier finden wir die idealen Voraussetzungen, damit der digitale Raum von hier aus „erobert“ werden kann.

Das Ziel der Region ist nicht so sehr, dass man nur das eine oder andere Projekt fördert, es geht schlicht um das Schaffen von Voraussetzungen für erfolgreiche Open-Commons-Projekte. Es braucht dann aber auch die sogenannten Leuchtturmprojekte, damit wir entsprechend sichtbar sind. Da wird es schon 2011 die ersten Ansätze im Bereich Open Government, Open Courseware oder Open Data geben.

Diese Inkubator-Funktion für neue Ideen, Communities und Unternehmen, egal ob kommerziell oder nicht kommerziell, will die Stadt Linz hier wahrnehmen. Dabei wird sie auch selber als Unternehmensgruppe Stadt Linz deutliche Akzente im Bereich Open Data und somit Open Government setzen. Das ist das Ziel dieser Open-Commons-Initiative. Wir haben in der Zwischenzeit auch mit Stefan Pawel einen hauptverantwortlichen Projektleiter eingesetzt, der all diese Aktivitäten koordiniert. In einigen Jahren wird man Referenzbeispiele und die eine oder andere neue Firma kennen, die sich in Linz unter der Open-Commons-Flagge entwickelt hat und national oder international an Bedeutung gewinnt. So wie jeder das Silicon Valley in Amerika kennt, wäre es ein sehr hohes Ziel, Linz im Bereich Open Commons europaweit und warum nicht auch weltweit bekannt zu machen.

Links

Public Space Server Linz

<http://pssinfo.public1.linz.at/>

Studie „Open Commons Region Linz“

http://www.freienetze.at/documents/ocr-studie/Studie_Open_Commons_Region_Linz.pdf

Unternehmensgruppe Stadt Linz

http://www.linz.at/politik_verwaltung/unternehmensgruppe.asp

Projekt: Freie Software verbreiten

Dass freie Software noch lange nicht gratis sein muss, ist *die* Binsenweisheit der Open-Source-Gemeinschaft. Auch wenn sich das Problem gleichen Zugangs zu aktueller Software für alle mit freier Software leichter lösen lässt, sind nicht alle aufgeworfenen Fragen dazu auch beantwortet. Was nützt beispielsweise freie Verfügbarkeit im Internet, wenn der Breitband-Internetzugang für den Download mehrerer GB an Daten unerschwinglich ist? Wie erschließen sich LaiInnen die Chancen und Vorteile freier Software, wenn sie am Beginn von der riesigen Vielfalt mehr verwirrt als beeindruckt werden? An diesen beiden Punkten lässt sich sehr einfach auch auf kommunaler Ebene ansetzen.

Zur Verringerung der finanziellen Hürden bei der Verwendung freier Software eignen sich „Freedom Toaster“¹⁴ an Bibliotheken und anderen öffentlichen Orten. Ausgestattet mit einem CD- oder DVD-Rohling, können an diesen Geräten, die aus einem Computer mit robustem DVD-Brenner bestehen („Toaster“ ist ja die englische Bezeichnung für CD- und/oder DVD-Brenner), sehr einfach CDs und DVDs mit freier Software erstellt werden. Größte Herausforderung ist denn auch weniger die Bereitstellung der Infrastruktur als vielmehr die übersichtliche Darstellung und kontinuierliche Wartung des Downloadangebots.

Als Ergänzung und gleichzeitige Vermarktung dieses Angebots bieten sich Linux-Live-DVDs – also DVDs, die freie und Open Source Software ohne Installation erlebbar machen – an, die speziell für bestimmte Zielgruppen maßgeschneidert sind. Derartige Systeme ermöglichen das einfache und ungefährliche Testen freier Software ohne Installation auch auf Windows-Computern. Die Stadt Wien unterstützt beispielsweise ein derartiges Pro-

¹⁴ Der Name und die Idee des „Freedom Toaster“-Projekts gehen auf die Shuttleworth-Stiftung des Ubuntu-Gründers und Millionärs Marc Shuttleworth zurück, der zahlreiche Freedom Toaster in Südafrika finanziert.

jekt mit dem Namen „Jux“ (<http://jux.netbridge.at>), das sich speziell an Kinder und Jugendliche wendet. Ähnliche Projekte für SeniorInnen, Kleinunternehmen oder den Bildungsbereich sind ebenfalls denkbar und wären mit Hilfe von entsprechenden KooperationspartnerInnen auch leicht umsetzbar.

Projektziele

- Schaffen von Anreizen für und Erleichterung von Verwendung freier Software
- Senkung der Einstiegshürden für spezielle Zielgruppen

Projektbestandteile

- Einrichtung/Anmietung und Wartung eines zentralen Servers, der Freie und Open Source Software für die gängigsten Anwendungsgebiete in aktueller Version bereithält
- Einrichtung und Wartung von „Freedom Toastern“, d. h. PCs mit CD/DVD-Brenner und einfachem Interface zur Erstellung von Datenträgern mit freier Software, mit Zugriff auf diesen Server in sämtlichen Bibliotheken und ausgewählten Volkshäusern zur Erstellung von CDs/DVDs mit freier Software für Menschen ohne (High-Speed-) Internetzugang.
- Entwicklung von zielgruppenspezifisch aufbereiteten Linux-Live-DVDs in Kooperation mit entsprechenden Zielgruppenorganisationen

Projektzielgruppe

- Völlige Neulinge im Umgang mit Freier Software jeden Alters, denen ein Ausprobieren so leicht wie möglich gemacht werden soll
- Sozial schwache SoftwarenutzerInnen, die nicht über Breitbandinternet und/oder mobile Computer für öffentliche Hotspots verfügen

ProjektträgerInnen

- Kommune, ggf. in Kooperation mit Zielgruppenorganisationen

Dialoggruppen

- Bibliotheken und Volkshäuser (für die Aufstellung der „Freedom Toaster“)
- Zielgruppenorganisationen (z.B. SeniorInnen- und Wirtschaftsverbände, etc.)

Mögliche Erweiterungen

- Kontinuierliche Erschließung weiterer Zielgruppen mit maßgeschneiderten Angeboten

Realisierungsbeispiel

- Im Bereich der Kinder- und Jugendarbeit das von der Stadt Wien unterstützte Projekt „Jux“, vgl. <http://jux.netbridge.at>

Projekt: Freie Software in der Kommunalverwaltung

Alle Stadtverwaltungen, die wie Wien oder München in jüngster Zeit verstärkt auf freie Software setzen wollten, standen beim Wechsel von Windows vor erheblichen Problemen. Und zwar steckten diese weniger in großen Fragen wie Funktionalität oder Umschulungsaufwand als vielmehr in versteckten Details. Unzählige kleine und kleinste Fachanwendungen arbeiten beispielsweise ausschließlich mit Microsoft Word zusammen. Verzeifelt berichten AdministratorInnen sogar von Webanwendungen, die eigentlich mit jedem beliebigen Internet-Browser funktionieren müssten und dennoch auf Microsofts Internet Explorer bestehen.

Ebenso erzählen IT-Verantwortliche, dass die Einführung von PCs genauso wie die Verwendung von Windows „passiert“ sei. Was beim Aufkommen völlig neuer Technologien noch entschuldbar sein mag, gilt sicher nicht im Wiederholungsfall. Der Ausweg aus der (teuren) Abhängigkeit von einem Hersteller erfordert denn auch einen langen Atem und strategische Planung. „Plattformunabhängigkeit“ lautet in diesem Zusammenhang das Gebot der Stunde. Offene Schnittstellen, Standards und Dateiformate, Webanwendungen für alle neuen Dienste und Leistungen sowie Ausschreibungen, die AnbieterInnen freier Software nicht benachteiligen, sind der Weg zur Erfüllung dieses Gebots.

Konkret bedeutet das die eingehende und transparente Beschäftigung mit der Softwareverwendung, sowohl in der kommunalen Verwaltung als auch in anderen Bereichen wie kommunalen Betrieben und Dienstleistern. So gibt es gerade im Gesundheitsbereich inzwischen zahlreiche Open-Source-Lösungen, die längerfristige Herstellerunabhängigkeit möglich machen. All das fordert aber zuerst ein deutliches politisches Bekenntnis zum übergeordneten Ziel der Plattformunabhängigkeit, das im Zweifel Entscheidungen

gegen kurzfristige Kostenvorteile mit langfristiger Abhängigkeit und damit für die Freiheit von Software und AnwenderInnen bedeutet.

Projektziele

- Plattformunabhängigkeit im Softwarebereich der kommunalen Verwaltung und Betriebe
- Förderung freier Software: Im Zweifel für die Freiheit

Projektbestandteile

- Erstellung einer Studie über die EDV-Situation in der Kommunalverwaltung sowie in kommunalen Betrieben mit dem Ziel, eine Roadmap zur Erreichung des Ziels „Plattformunabhängigkeit“ zu erstellen
- Verfassung einer mittel- und langfristigen IT-Strategie mit verbindlichen Kriterien für künftige Ausschreibungen und Beschaffungsvorgänge.

Projektzielgruppe

- Die politischen und administrativen IT-Verantwortlichen in der Kommunalverwaltung und den kommunalen Betrieben
- Mittelbar die KooperationspartnerInnen sowie die IT-Bezugsquellen

ProjekträgerInnen

- IT-Abteilung bzw. öffentliche IT-DienstleisterInnen, evtl. Fremdvergabe der Studie

Dialoggruppen

- IT-Verwaltung
- Politik
- IT-Verantwortliche in kommunalen Betrieben

Finanzierungsbedarf

- Kein übermäßiger Finanzierungsbedarf, allenfalls Anschubfinanzierung zur Erstellung der Studie

Mögliche Erweiterungen

- Freigabe von Auftragsentwicklungen zur Weiternutzung von anderen Stadtverwaltungen

Realisierungsbeispiel

- Das umfassendste Umstellungsbeispiel einer Stadtverwaltung ist das Münchner LiMux-Beispiel: <http://www.muenchen.de/limux>

Projekt: Freie Software für Schulen

Auf den ersten Blick mag es noch sinnvoll erscheinen, an Schulen die „gängigste“ Software einzusetzen und zu unterrichten. Auf den zweiten Blick überwiegen jedoch die Nachteile der gegenwärtigen Microsoft-Monokultur: SchülerInnen erlernen nicht den Umgang mit einer Technologie, sondern nur den mit einem Produkt. Die günstige Ausstattung der Schulen mit Microsoft-Software ist vor diesem Hintergrund auch alles andere als selbstloses oder soziales Engagement – ganze Generationen von SchülerInnen werden so zu künftigen Microsoft-KundInnen herangezogen. Außerdem ist es bei proprietärer Software nicht möglich, den SchülerInnen diese einfach auch für den Einsatz zu Hause mitzugeben: In der Regel müssen die Eltern dafür (tief) in die Tasche greifen.

In Anlehnung an erfolgreiche Open-Source-Projekte im Bildungsbereich wie die Linux-Distribution des Schleswig-Holsteiner Bildungsamts (*kmLinux*) oder *Edubuntu* können maßgeschneiderte DVDs für die Verwendung an Schulen zusammengestellt werden. Sie enthalten nicht nur ein freies schülerInnenfreundliches Betriebssystem, sondern auch eine Vielzahl an pädagogischen Programmen. Die Auswahl ist gigantisch: Von Vokabeltrainern über Erkundungssoftware zum Periodensystem bis hin zur Geometrie wird jedes Fachgebiet abgedeckt. Selbst eine Programmiersprache für Kinder wurde entwickelt. Auch für LehrerInnen ist etwas dabei, z. B. Programme zur Zusammenstellung und Auswertung von Tests und Verwaltung von SchülerInnenendaten und Leistungen. Anfangs sollten zumindest alle LehrerInnen mit der DVD ausgestattet werden, mittelfristig können auch alle SchülerInnen einbezogen werden. Da die Software frei ist, dürfen aber ohnehin jederzeit und für jeden Zweck auch in der Schule Kopien angefertigt werden.

Interessierte Lehrkräfte hätten über das Internet bereits heute Zugang zu diesen Programmen. Die Bereitstellung der Software alleine ist deshalb noch

nicht genug, um für mehr Vielfalt in schulischen Softwarelandschaften zu sorgen und möglichst vielen LehrerInnen und SchülerInnen die Chancen freier Software zu eröffnen. Deswegen sollten begleitende Fortbildungskurse für LehrerInnen an entsprechenden Fortbildungseinrichtungen angeboten und beworben werden. Dabei soll zwischen den unterschiedlichen Fachbereichen (Software für den Einsatz in Volksschulen, in der Sekundarstufe oder im Informatikunterricht) unterschieden werden. Ziel ist es, diese Kurse über den offiziellen Fortbildungstopf für LehrerInnen anzubieten, um möglichst viele Lehrkräfte zum Einsatz von Freier Software zu befähigen und motivieren.

Projektziele

- Unterstützung von LehrerInnen beim Einsatz von freier Software im Unterricht
- Schaffung einer größeren Vielfalt in der schulischen Softwarelandschaft

Projektbestandteile

- Entwicklung eines Fortbildungskurses für interessierte LehrerInnen an der jeweiligen Fortbildungseinrichtung
- Distribution einer DVD mit freier Software, abgestimmt auf den Einsatz im Unterricht und zu Hause

Projektzielgruppe

- Interessierte LehrerInnen verschiedener Schultypen und Unterrichtsgegenstände sowie angehende LehrerInnen
- SchulleiterInnen

ProjekträgerInnen

- Fortbildungseinrichtungen
- Lokale Schulverwaltungsbehörden

Dialoggruppen

- Fortbildungseinrichtungen
- Schulverwaltungsbehörden
- IT-Verantwortliche und (EDV-)LehrerInnen

Finanzierungsbedarf

- Anlaufkosten: Entwicklung eines Kurs-Konzeptes, Erstellen von Kursunterlagen, Auswahl der Software
- Laufende Kosten: Abhalten der Kurse, Vervielfältigung der DVDs

Mögliche Erweiterungen

- Neben Schulen bieten sich auch andere Bildungseinrichtungen wie Kindergärten und Vorschulen für derartige Initiativen an.

Realisierungsbeispiel

- Das Projekt *kmLinux* des Landesbildungsservice Schleswig-Holstein: <http://www.lernnetz-sh.de/kmLinux/>

Bildnachweise

- Leonhard Dobusch (Cover)
Foto (Ausschnitt) von Joi Ito
- Christian Forsterleitner (Cover)
Foto (Ausschnitt) von Joi Ito
- Manuela Hiesmair (Cover)
Foto (Ausschnitt) von Rubra
- Karin Frohner (S. 25)
Foto (Ausschnitt) Stadtkommunikation Linz
- Volker Grassmuck (S. 29)
Foto (Ausschnitt) Raimond Spekking (CC by-sa)
- Lawrence Lessig (S. 61)
Foto (Ausschnitt) von Joi Ito
- Gerda Forstner (S. 65)
Foto (Ausschnitt) von Oliver Theusl (CC by-sa)
- Anne Margulies (S. 91)
Foto (Ausschnitt) Stephanie Mitchell, Harvard Staff Photographer
- Monika Andraschko und Thomas Lumplecker (S. 97)
Foto (Ausschnitt) Barbara Hofmann
- Richard M. Stallman (S. 123)
Foto (Ausschnitt) Bill Ebbesen
- Gerald Kempinger (S. 127)
Foto (Ausschnitt) IKT Linz GmbH (CC by-sa)
- Markus Beckedahl (S. 157)
Foto (Ausschnitt) Euphoriefetzen (CC by-sa)
- Gregor Kratochwill (S. 161)
Foto (Ausschnitt) Kratochwill (CC by-sa)

- Gerhard Fröhlich (S. 185)
Foto (Ausschnitt) Christo Christoph
- Melissa Hagemann (S. 191)
Foto (Ausschnitt) Beatrice Murch (CC by-sa)
- Wendy Hall (S. 213)
Foto (Ausschnitt) University of Southampton (CC by-sa)
- Johann Höller (S. 219)
Foto (Ausschnitt) idv (CC by-sa)
- David Eaves (S. 247)
Foto (Ausschnitt) Eaves (CC by-sa)
- Gustav Pomberger (S. 251)
Foto (Ausschnitt) Johannes Kepler Universität Linz