
Auswirkungen von Open-Source-Software auf den Arbeitsmarkt

Eine Übersichtsstudie im Auftrag des
Bundesministeriums für Wirtschaft und
Arbeit

Verfasst von Reinhard Gantar, Francisco Webber, Nikolaus Kimla, Gabriele Brötzner

Juni 2003

Inhaltsverzeichnis

1.1.	ZIELSETZUNGEN.....	4
1.1.1.	Ausgangsthesen	4
1.1.2.	Vorgangsweise.....	5
2.	VORBETRACHTUNGEN	7
2.1.	Fachkräftemangel in der IT-Branche.....	7
2.2.	Fragen der strategischen Ausrichtung beim Einsatz von Informationstechnologie.....	10
2.3.	Ist die IT-Branche ein einheitlicher Arbeitsmarkt?	12
2.4.	IT-Kostenfaktoren bei KMU.....	14
2.5.	Bewertung der Ergebnisse der Software-Studie 2000 im Zusammenhang mit Open-Source-Software	15
3.1.	EINSATZGEBIETE VON OPEN-SOURCE-SOFTWARE	17
3.2.	METRIKEN FÜR WIRTSCHAFTLICHKEIT UND QUALITÄTSSICHERUNG VON OS.....	18
3.3.	ZUSAMMENHANG VON TECHNIK UND GESELLSCHAFT IM BEZUG ZUM OPEN-SOURCE-KONZEPT. KREATIVES UND PRODUKTIVES POTENTIAL DER OS ENTWICKLERGEMEINDE.....	20
3.4.	STANDARDS UND INTEROPERABILITÄT VON OPEN-SOURCE- SOFTWARE.....	22
4.1.	DIE ZUNAHME VON OPEN-SOURCE-SOFTWARE IN DER WIRTSCHAFT FÜHRT ZU EINEM WACHSENDEN BEDARF AN OS-FACHKRÄFTEN.....	24
4.2.	STETIGE DIVERSIFIZIERUNG UND SPEZIALISIERUNG FÜHRT ZUM ENTSTEHEN NEUER BERUFE UND BERUFSZWEIGE....	26
4.3.	DAS ANFORDERUNGSPROFIL AN OS-FACHKRÄFTE (<i>KREATIVITÄT, MOTIVATION, TEAMWORK, MEDIATOR</i>).....	27
4.4.	FACHKRÄFTE UND TELEARBEIT, NEUE SELBSTÄNDIGE, MICRO-ENTERPRISES, COMPETENCE-NETWORKS	29
4.5.	LISTE DER AM HÄUFIGSTEN EINGESETZTEN OS-SOFTWARE IM WIRTSCHAFTLICHEN UMFELD	31
4.5.1.	Das Betriebssystem Linux.....	31
4.5.2.	Die *BSD Familie	31
4.5.3.	GNU Software.....	31
4.5.4.	Der Serverdienst Sendmail	31
4.5.5.	Der Emacs Editor.....	32

4.5.6.	Der Vim Editor	32
4.5.7.	Der Apache Webserver	32
4.5.8.	Die Datenbanksysteme PostgreSQL, MySQL	32
4.5.9.	Die Skriptingsprache Perl	32
4.5.10.	Die Webprogrammiersprache PHP	32
4.5.11.	Die Skripting Umgebung Python	33
4.5.12.	Samba.....	33
4.5.13.	Der Applikationsserver Zope	33
4.5.14.	Der Linux-Desktop KDE	33
4.5.15.	Der Linux-Desktop GNOME.....	33
4.5.16.	Das UNIX Windowingsystem Xfree86	34
4.5.17.	Der Serverdienst BIND	34
4.5.18.	Der Serverdienst INN	34
4.5.19.	Das Textsystem TeTex.....	34
4.5.20.	Das 3D Rendering Package Persistence of Vision	34
4.5.21.	OpenOffice (vormals StarOffice).....	34
4.5.22.	Abiword	35
4.5.23.	KOffice	35
4.5.24.	Mozilla	35
5.1.	DIE AUSBILDUNG VON OS-FACHKRÄFTEN UND „LEBENSLANGES LERNEN“	36
5.2.	DIVERSIFIZIERUNG DER AUSBILDUNGSMÖGLICHKEITEN	36
5.3.	OPTIONEN EINES RESTRUKTURIERTEN ANGEBOTES DER ETABLIERTEN BILDUNGSTRÄGER: UNIVERSITÄTEN, FACHHOCHSCHULEN, HÖHERE SCHULEN, BERUFSSCHULEN, ERWACHSENENBILDUNG.....	37
5.4.	ZERTIFIZIERUNGSSYSTEME FÜR OSS.....	38
5.5.	AUS DER PRAXIS: BEISPIELE FÜR NEUE BERUFSBEZEICHNUNGEN	38
5.5.1.	LAMP-Experte	38
5.5.2.	Zopist.....	38
5.5.3.	Markup-Specialist/XML-Designer	38
5.5.4.	CMS-Entwickler.....	39
5.5.5.	Community Liasons.....	39
5.5.6.	Open-Source-Dokumentation	39

6.1.	EINSATZMÖGLICHKEITEN VON OS-SOFTWARE AUßERHALB DES AMS AN ÖFFENTLICHEN „ACCESS POINTS“ ODER ALS SPEZIELLE CLIENT SOFTWARE	40
6.2.	KLÄRUNG DER VORAUSSETZUNGEN ZUR OPTIMIERUNG EINES OFFENEN DATENTRANSFERS (AMS - PRIVATE BETRIEBE) 40	
7.1.	POSITIONIERUNG DES THEMAS IM RAHMEN DES EU PROGRAMMS E-EUROPE	42
7.2.	MÖGLICHKEITEN DES EINSATZES VON OS-SOFTWARE IM BEREICH DER ÖFFENTLICHEN VERWALTUNG	42
7.3.	ERFAHRUNGEN UND STRATEGIEN VON OS-SOFTWARE IM ÖFFENTLICHEN BEREICH IN DEUTSCHLAND, FRANKREICH, SPANIEN, ENGLAND, BELGIEN.....	43
7.4.	EIN KURZER AUFRISS DER UNTERSCHIEDE IM OS ENTWICKLUNGSBEREICH USA - EU - CHINA/TAIWAN	44
7.5.	LOKALE WERTSCHÖPFUNG UND AUSWIRKUNGEN AUF DIE PRODUKTIVITÄT	47
8.1.	ARBEITSMARKT UND BILDUNG	50
8.2.	DIE WIRTSCHAFT.....	51
8.3.	DIE ÖFFENTLICHE HAND	52
	ANHANG 1: REFERENZEN	53
	Allgemein	53
	OSS in China/Taiwan.....	56
	ANHANG 2: GLOSSAR	58
	ANHANG 3: INDEX	69

Open-Source-Software im Kontext des Arbeitsmarktes

Eine Übersichtsarbeit zum Thema:

1.1. Zielsetzungen

Innerhalb der EU wird der Einsatz von Open-Source-Software in der Verwaltung und die Förderung ihrer Verbreitung in der Privatwirtschaft rege diskutiert. Open-Source-Software hat in diesen Bereichen einige markante Vorzüge, ist aber hinsichtlich ihrer Auswirkung auf den Arbeitsmarkt und die Wirtschaft noch nicht genauer untersucht worden.

Diese Arbeit soll die Thematische Ausgangsposition beschreiben und Grundlage für weitere (auch empirische) Studien dienen. Die inhaltlichen Bereiche lassen sich wie folgt umreißen:

- 1) *Absehbare Auswirkung von Open-Source-Software auf den Arbeitsmarkt .*
- 2) *Anforderungen an die Aus- und Weiterbildung.*
- 3) *Volkswirtschaftliche Folgeabschätzungen und Potentiale.*
- 4) *Europäische Kooperation beim Einsatz und der Entwicklung von Open-Source-Software in der öffentlichen Verwaltung.*
- 5) *Erkenntnisse zum Themenkreis Open-Source-Software und Gesellschaft innerhalb Europas*

1.1.1. Ausgangsthesen

Nicht alle Einsatzgebiete sind für Open-Source-Software gleich gut geeignet, jedes Einsatzgebiet muss aus der Perspektive von Open-Source-Entwicklern (Firmen und Individuen) und Anwendern (Individuen, Unternehmen und der öffentlichen Verwaltung) beleuchtet werden. Grundsätzlich liefern die bisherigen Erfahrungen mit Open-Source-Software Anlass zur Hoffnung, dass sie nicht nur geringere Kosten verursacht, sondern EU-weit die Entwicklung von wichtigen interoperativen Standards über die Grenzen der einzelnen Mitgliedstaaten hinweg beschleunigt. Weiters liefert die Pflege und freie Zirkulation von Open-Source-Software vermutlich wichtige Anreize zur eigenständigen Forschung und Entwicklung von Software innerhalb der EU. Die Förderung von Open-Source-Software kann Vorteile im Standortwettbewerb bringen und Arbeitsplätze in einer breit gefächerten regionalen Software- und Telekommunikationsindustrie schaffen. Ob und wie weit diese Thesen stichhaltig sind, gilt es in dieser Studie zu diskutieren.

1.1.2. Vorgangsweise

1.1.2.1. Bildung der Steuergruppe

Das gesamte Studienprojekt wurde von einer Steuergruppe unterstützt. Namhafte Experten aus Wirtschaft, Universität und Verwaltung diskutieren dabei synergetisch den Fortgang. Ein Projektteam der Universität Klagenfurt etwa nahm sich dem Thema aus volkswirtschaftlicher Sicht an, Vertreter der Wirtschaft beleuchteten praxisnahe und rechtliche Aspekte, das IT-Board der Bundesregierung argumentierte von politischer Seite her.

Zusätzlich wurden im Rahmen des Studienprojekts bei einer öffentlichen Veranstaltung für Vertreter österreichischer KMU grundsätzliche, praktische und rechtliche Aspekte beim Einsatz von Open-Source-Software diskutiert. Geladen hatte dazu die Wirtschaftskammer Österreich in Kooperation mit dem BMWA.

Zur Steuergruppe zählen folgende Personen:

<i>Gabriele Brötzner,</i>	<i>uptime Systemlösungen</i>
<i>Herr Dirnbauer,</i>	<i>web-consulting</i>
<i>Andreas Eckwolf,</i>	<i>BMWA</i>
<i>Reinhard Gantar,</i>	<i>uptime Systemlösungen</i>
<i>Christa Giffinger,</i>	<i>BMWA</i>
<i>Gottfried Haber¹,</i>	<i>Universität Klagenfurt</i>
<i>Ronald Hechenberger,</i>	<i>oiat</i>
<i>Nikolaus Kimla,</i>	<i>uptime Systemlösungen</i>
<i>Markus Kleemann,</i>	<i>Eurojobs</i>
<i>Friedrich Kofler,</i>	<i>WKÖ</i>
<i>Robert Kromer,</i>	<i>Kromer IT Consulting</i>
<i>Wolfgang Neurath,</i>	<i>BMWA</i>
<i>Christian Operschall,</i>	<i>BMWA, Vorsitz</i>
<i>Reinhard Posch,</i>	<i>Bundeskanzleramt</i>
<i>Francisco Webber,</i>	<i>uptime Systemlösungen</i>

¹ Im Zuge der Entwicklung der vorliegenden OSS-Studie von uptime nahmen auch Wirtschaftsexperten der Universität Klagenfurt, die derzeit an einer Studie über die gesamtwirtschaftlichen Effekte des Software-Sektors in Österreich, an der Steuerungsgruppe teil. Zur Illustration werden hier auch > einige Kommentare des Forscherteams zu einzelnen Punkten der Studie wiedergegeben, die als weiterführende Diskussionsanregungen zu den vielen wertvollen Inhalten und Denkanstößen der vorliegenden Arbeit dienen sollen. Trotz einer engen Zusammenarbeit und zahlreicher beiderseits fruchtbarer Diskussionen, liegt das Verdienst der geistigen Urheberschaft der Studie selbst vollständig bei uptime, während für die Inhalte der Kommentare ausschließlich das Forscherteam der Universität Klagenfurt verantwortlich ist."

1.1.2.2. Workshops

Zur Erstellung der Studie traf sich die Gruppe der Workshopteilnehmer regelmäßig seit der Auftragserteilung des BMWA von Dezember 2002 bis Ende Mai 2003.

Workshopteilnehmer waren: *Nikolaus Kimla*
Francisco Webber (Diskussionsleitung)
Reinhard Gantar
Gabriele Brötzner (Schriftführung)

1.1.2.3. Redaktion

uptime Systemlösungen GmbH
Schwarzenbergplatz 8/8, 1030 Wien

2. Vorbetrachtungen

2.1. Fachkräftemangel in der IT-Branche

2.1.1. Angebot und Nachfrage an IT-Fachkräften

Weder die Boom-Jahre des Internet (ca. 1998-2001) noch die darauf folgende Konjunkturschwäche sollten davon ablenken, dass der Einsatz von Informations- und Telekommunikationstechnik - langfristig gesehen - weiter steigen wird. Weder der Engpass an Internet-versierten Fachleuten während des Booms noch das augenblickliche Überangebot an diesen Spezialisten sind typische Erscheinungen.

Zum einen betraf die „dot.com-Bubble“ vorwiegend zwei enge Segmente der gesamten IT-Branche (e-Commerce und Mobiltelefonie, bzw. deren Technologie-Lieferanten), zum anderen entbehrten die zugrundeliegenden Geschäftsmodelle jeder betriebswirtschaftlichen Substanz. Die Unternehmen vergeudeten auch gewaltige Ressourcen durch Einsatz damals noch sehr unzulänglicher und unerprobter Mittel, weil die technischen Voraussetzungen für die angestrebten Ziele erst noch geschaffen werden mussten. Es war in der Pionierzeit spezialisierter Web-Services (Content, e-Commerce) sehr viel schwieriger als heute, Datenbanken, Web-Pages und Internet-Funktionen zu integrieren. Die vielfältigen Bibliotheken, Software-Werkzeuge, Know-how, etc. gab es damals noch nicht, ebenso wenig wie etablierte Kriterien, etwa die Fähigkeiten von Fachkräften zu bewerten sind.

Studien, Überlegungen und Beobachtungen aus dem damaligen Feld des IT-Arbeitsmarktes müssen daher sehr sorgfältig geprüft werden, inwieweit das Material für nüchternere Zeiten aussagekräftig ist.

Man kann den Ist-Zustand des Arbeitsmarktes natürlich nicht unabhängig vom Ist-Zustand des Marktes selbst untersuchen. Daher sollte man im Auge behalten, dass nur wenige Unternehmen selbst IT-Leistungen anbieten, obwohl sie selbst (oft großen) Bedarf nach IT-Fachkräften haben und für den eigenen Gebrauch umfangreiche Software und Services schaffen. Wie viel oder wie wenig sich ein Unternehmen diese Leistungen kosten lässt, und wie kritisch sich deren Qualität auf die Geschäftsabwicklung auswirkt, ist von Branche zu Branche und sogar von Unternehmen zu Unternehmen verschieden. Generell gilt, dass die Bedeutung von internen und externen IT-Dienstleistungen langfristig zunehmen und in vielen Fällen eine wettbewerbsentscheidende Ressource werden wird. Für die Zukunft ist daher auch eine größere Nachfrage an IT-Fachkräften zu erwarten.

2.1.2. Zukünftige Entwicklung des IT-Arbeitsmarktes

Trotz seiner immer noch enormen Präsenz in den Medien und dem Bewusstsein der Öffentlichkeit sind das Internet und e-Commerce nicht die typischen IT-Anwendungen im Unternehmen. Zwar wird die Bedeutung und der Anteil an Gütern und Dienstleistungen steigen, die über das Internet angeboten und vertrieben werden, wesentlich mehr IT-Fachkräfte aber bleiben in den klassischen Sektoren der IT für interne Verwaltung und Produktion beschäftigt. Sogar der Trend zum Einsatz von Web-Anbindungen an die klassische und erprobte interne IT-Leistungen über das Inter-, Intra- oder Extranet wird nur einen verhältnismäßig kleinen Teil des IT-Arbeitsmarktes betreffen. Andererseits werden erwartungsgemäß sehr viele Arbeitnehmer mit diesen Applikationen als Benutzer in Berührung kommen.

Inwiefern sich Angebot und Nachfrage von passend qualifizierten IT-Fachkräften decken, war nie einfach zu untersuchen oder vorherzusehen, denn beides hängt unter anderem von folgenden Einflüssen ab, die selbst teilweise dem gängigen Modus Operandi und den strategischen Planungen konkurrierender Unternehmen unterliegen.

1.) Die Palette an Programmiersprachen, Ansätzen, Technologien und Methodologien ist bereits heute für den Einzelnen unüberschaubar und wird in Zukunft noch umfangreicher und facettenreicher werden. Individuelle Kenntnisse einer bestimmten Programmiersprache oder Technologie bedeuten nur für verwandte Teilbereiche eine einfache Umschulung. Viele Teilbereiche sind konzeptionell so weit voneinander entfernt, dass ein vollständiger Umstieg für den Betroffenen Jahre dauern könnte. Leider entwickeln sich IT-Technologien und/oder deren Nachfrage in der Industrie oft rascher als es für den einzelnen Menschen oder für ein Bildungsinstitut vorhersehbar oder nachvollziehbar ist. (Beispiele: Java, XML, SOAP, Renaissance der Mainframes, etc.)

2.) Wie die Kosten für die Umschulung eines individuellen Arbeitnehmers vom Arbeitgeber, von der Gesellschaft und vom Betroffenen selbst aufgeteilt werden sollen, bis zu welchem Alter er als Umsteiger für seinen bestehenden oder potentiellen Arbeitgeber glaubwürdig ist, und wie lange diese Umschulung oder Neuqualifizierung dauert, ob sie berufsbegleitend, vorbereitend oder in Sabbaticals erworben werden soll, das alles unterliegt teilweise einem schnell wechselnden Konsens, hängt von der Qualität der angewendeten Methoden und der allgemeinen Wirtschaftslage ab.

3.) Der Einsatz oder Nichteinsatz bestimmter technologischer oder organisatorischer Ansätze ist nicht für alle Segmente gleich sinnvoll oder durchschlagskräftig. Der Marktwert einer entsprechenden Qualifizierung bzw. die Rentabilität einer Investition ist weder für das Individuum (etwa einen Programmierer oder Systemadministrator) noch den Entscheidungsträger im Unternehmen ohne weiteres absehbar. Egal, ob Analysten, Manager, Studenten, Consultants, Personalentwickler - niemand ist davor gefeit, die Verbreitung einer bestimmten Technologie (im weitesten Sinne) zu über- oder unterschätzen. Die damit verbundenen

Opportunitätskosten können sowohl für Unternehmen als auch für das Individuum beträchtliche Auswirkungen haben.

All das bedeutet, dass ein Reservoir an formal hervorragend qualifizierten IT-Fachkräften nicht notwendigerweise die gesamte Nachfrage decken wird, auch wenn beides rein numerisch ausgeglichen ist. Wenn eine Qualifizierung am Markt vorbei geht, wenn sich die (in manchen Fällen umstrittenen) Schwerpunkte der Industrie nicht mit den Ausbildungsschwerpunkten der Individuen decken, kann gleichzeitig ein Überangebot UND ein Mangel an IT-Fachkräften entstehen. Open-Source-Software kann wegen seiner prinzipiell und traditionell sehr langen Lebenszyklen helfen, solchen Erscheinungen vorzubeugen (siehe Abschnitt 3.3).²

Diese Beobachtungen haben auch Open-Source-spezifische Auswirkungen auf die Lebensarbeitszeit von Fachkräften (siehe Abschnitt 3.3). Das Bild vom älteren Arbeitnehmer, der nicht mehr imstande ist, sich an neue technische Entwicklungen anzupassen, stimmt für die IT-Branche und ihre Fachkräfte vermutlich weniger als für andere Branchen. Trotzdem läuft jeder Programmierer, Administrator und Projektleiter Gefahr, eines Tages mit Kenntnissen dazustehen, für die der Markt zu klein geworden ist. Diese Bedrohung ist wegen des Beharrungsvermögens der IT-Nutzung in der Praxis weniger dramatisch als gemeinhin angenommen. Da Technologien ihre Blütezeit oft um Jahrzehnte überleben, ist es meist so, dass mit dem Überschreiten des Zeniths eher der erforderliche Nachschub an Fachkräften fehlt (Beispiele: Novell, Delphi, besonders eklatant: Großrechner) und zu einem Engpass beim Angebot führt.

2.1.3. Auswirkungen der Konjunktorentwicklung

Da anteilmäßig nur wenige Unternehmen IT-Dienstleistungen verkaufen und diese daher nur als Kostenfaktor sehen können, gilt für IT-Ausgaben alles, was sich auf über andere Investitionen abseits der unmittelbar angebotenen Güter oder Dienstleistungen sagen lässt: wie im Bereich des Marketing oder der Personalentwicklung führen auch bei der IT-Infrastruktur mangelnde Einnahmen typischerweise zu einem Rückgang der Investitionen und führen zu konservativeren, auf kurzfristige Einsparung konzentrierte Entscheidungen. Für den Ist-Markt und für den Ist-Arbeitsmarkt bedeutet das: risikoreichere Projekte werden verschoben oder verworfen, viele kurzfristig teure, aber langfristig rentable Investitionen in Infrastruktur unterbleiben, die Budgets für Weiterbildung sinken. Vielen Unternehmen kostet die Krise ihre Existenz.³ Es fehlen differenzierte Untersuchungen

² Forscherteams der Universität Klagenfurt (FdUK); „Die Länge der Lebenszyklen einzelner Software-Versionen ist im OSS-Bereich in vielen Fällen deutlich kleiner, allerdings bleiben „alte“ Technologien in der Tat relativ lang verfügbar.“

³ FdUK; „Dieser Effekt (Einsparung bei strukturellen Investitionen) in Phasen geringen Wirtschaftswachstums lässt sich empirisch bestätigen. Softwareinvestitionen sind jedoch, wie hier beschrieben, generell wichtige Wachstumsfaktoren für Unternehmen.“

darüber, welche spezifischen IT-Kenntnisse und Qualifikation vom Marktgeschehen besonders berührt werden, so wie darüber welche und wie viele Spezialisten Probleme bei der Integration in den Arbeitsmarkt haben.

2.2. Fragen der strategischen Ausrichtung beim Einsatz von Informationstechnologie

2.2.1. Strategische Faktoren beim Einsatz von Informations-Technologie

Wie in 1.2 erwähnt, hängen die Bedeutung und die Details der IT-Nutzung von der jeweiligen Branche ab, und welche Rolle der IT typischerweise in ihr gegeben wird. Grob unterteilen kann man aber in

a.) logistischen Einsatz (eher Güter-orientierte Unternehmen) und
b.) infrastrukturellen Einsatz (eher Service-orientierte Unternehmen).
Diese Unterteilung gilt für „klassische“ Unternehmen, die keine IT-Services oder Software verkaufen, trotzdem entfaltet Open-Source-Software auch hier ihre Vorzüge (laut FLOSS, Teil 2, p. 47, Abschnitt 5.1.5, vier Argumente, die für den Einsatz von Open-Source-Software sprechen):

- 1.) *Höhere Ausfall- und Einbruchssicherheit (Stability and Security) durch eine große Zahl an Freiwilligen, die den Code untersuchen, testen und warten.*
- 2.) *Geringe oder gar keine Lizenzgebühren.*
- 3.) *Mehr Flexibilität bei Konfiguration, Integration der Produkte untereinander, gegebenenfalls können Leistungsmerkmale hinzugefügt werden.*

2.1.2. Bedeutung dieser Faktoren

Wie viel die IT-Infrastruktur zur Rationalisierung der Prozesse beiträgt, hängt von der Qualität der Planung und Implementation, und damit von den Entscheidungsträgern ab. Da aber das Maß an Einsparung, Ausfallsicherheit, Benutzerergonomie, etc. wettbewerbsentscheidend sein kann und - im Gegensatz zu anderen ökonomischen Faktoren - noch Potential bietet, wird sich der Wettbewerb in Zukunft auch bei solchen Unternehmen in Richtung IT verschieben, die selbst keine IT-Services oder Software anbieten. Die Bedeutung von IT als Produktionsmittel („Faktor“) wird also in allen Unternehmen steigen - wie diese Entwicklung etwa bei der Automobilindustrie schon vor längerer Zeit begonnen hat (Automatisierung, Computer-Integrated Manufacturing, virtual Prototyping, etc.)⁴ Diese Wichtigkeit von IT entsteht nicht nur durch die Verbilligung und Beschleunigung der internen Verwaltung und Automatisierung der Produktion, sondern auch durch die Integration der

⁴ FdUK; „Diese Einschätzung wird in der Literatur fast uneingeschränkt geteilt und muss unterstrichen werden.“

internen Prozesse mit denen der Lieferanten, Kunden (Business-to-Business) und Regierungsstellen über das Internet.

Erfahrungsgemäß nutzen eher Großunternehmen als KMU die Vorteile von Open-Source-Software. Dieser Umstand ergibt sich NICHT durch eine schlechtere Eignung von OSS für KMU, sondern erklärt sich durch die längere Tradition von IT überhaupt in großen Unternehmen - sie haben jahrzehntelange Erfahrung mit dem Einsatz von Informationstechnologie. Darunter sind zu einem bedeutenden Teil Betriebssysteme und Technologien, die den typischen OSS-Technologien sehr ähneln (Unix); KMU, auf der anderen Seite, sind typischerweise erst seit Mitte der 80er mit PC-kompatiblen Systemen mitgewachsen, die sich erst heute dank des technischen Fortschritts den Möglichkeiten der Groß-EDV annähern. Dem breiten Einsatz von OSS in KMU steht, zumindest auf dem Gebiet der Netzwerkinfrastruktur, lediglich ein Bewusstseinsbildungsprozess im Weg.

Bei Business Process- und Office Productivity Software sind Standardsicherheit und Benutzerergonomie zentrale Anliegen. Ob und wieweit hier von KMU OSS-Alternativen zu gängigen proprietären Systemen eingesetzt werden können, hängt von den konkreten Aufgaben ab, kann aber für den größten Teil des Terrains durchaus bejaht werden (Open Office, Star Office, diverse Datenbank-Produkte, etc.)

2.1.2.3. Ist IT eine „Güterindustrie“ oder ein Dienstleistungsgewerbe“?

Die Idee von Software als Dienstleistung ist nicht neu, wurde aber von Eric S. Raymond („Die Kathedrale und der Basar“) speziell im Zusammenhang mit Open-Source-Software neu aufgegriffen. Nach Raymond ist die gängige Vorstellung von Softwarepaketen, die in Fabriken abgepackt und von Händlern zu einem bestimmten Preis zum Kunden gebracht werden, von nur geringer Bedeutung und wurde erst mit der Verbreitung von Personal Computern üblich. Software als Ware wird wieder verschwinden, denn tatsächlich ist sie mehr ein Rohmaterial, das Installation, Wartung, Anpassung, Ergänzungen und Installation neuer Versionen - Service - erfordert. Da der fixe Preis für ein bestimmtes „Paket“ nur selten Kostenwahrheit zeigt - die einmalige Vergebühung verteilt die Kosten für Entwicklung, Vertrieb und Support mehr oder weniger gleichmäßig auf alle noch so ungleichen Käufer - ist ein Service-Modell angemessener, besonders bei komplexer Infrastruktur-Software, die regelmäßige Überprüfung und Administration durch Fachleute erfordert.⁵ Open-Source-Software begünstigt eine extreme Ausformung dieses Gedankens: Die Software selbst hat hier einen Verkaufspreis von Null, nur die Services werden für jeden Kunden gesondert vom Dienstleister verbucht. Gerade den großen Lieferanten von

⁵ FdUK; „Diese u.a. auf Raymond zurückgehende Ansicht über das Verschwinden von Software als Ware wird von einigen Ökonomen allerdings bezweifelt. Im Bereich der Standardsoftware wird auch langfristig ein Markt für ein „genormtes“ Produkt bestehen, egal ob der Quellcode offen ist oder nicht. Auch die Verteilung von direkten Kosten und nicht-direkten Kosten (z.B. für Forschung und Entwicklung) auf investitionsintensive Produkte ist betriebswirtschaftlich nicht ungewöhnlich (Pharma-Industrie, Telekom, Energie,...). Voraussichtlich wird aber der Dienstleistungsanteil weiter steigen ...“

IT ist diese Beobachtung nicht entgangen. IBM⁶ und SAP haben sich immer als Service-Unternehmen und IT-Betriebsberater gesehen, nicht als Software- oder Hardware-Hersteller. Inzwischen folgen sogar traditionell reine Hardware-Hersteller diesem Trend zur Dienstleistung.

2.3. Ist die IT-Branche ein einheitlicher Arbeitsmarkt?

2.3.1. Unterschiedliche Repräsentation der verschiedenen Sektoren im IT-Bereich.

Es gibt große Unterschiede in der Differenzierung des IT-Marktes, davon abgesehen gibt es zwei Arten von Schwerpunkten (IT-Zentren): Industriezentren und Universitäten; siehe nächster Abschnitt. Kooperation dieser Typen von Zentren gibt es so gut wie keine, wäre aber in jedem Fall wünschenswert. Die Durchschlagskraft der amerikanischen Forschung, Entwicklung geht nicht zu letzt auf die enge Verflechtung von staatlichen (militärischen) Subventionen für universitäre Forschung und anschließender Vermarktung der daraus entstehenden Produkte zurück.

2.3.2. Regionale Unterschiede in der IT-Branche

Obwohl Informationstechnologie von physischen Ressourcen unabhängig ist, konzentrieren sich die Lieferanten und Dienstleister in und um urbane Zentren. Das führt dazu, dass die Qualität der Lösungen typischerweise mit der Entfernung zu diesen Zentren abnimmt. Diese regional Inhomogenität gliedert sich weiter um staatliche Forschungsstätten (Universitäten), die offenbar ein für Open-Source-Software günstigeres Klima erzeugen als industrielle IT-Zentren. Es liegt auf der Hand, dass auch für die IT-Zentren die üblichen Prinzipien der Industrie- und Gewerbeansiedlung gelten: Konzentration wird Anreiz für noch mehr Konzentration. Einsatz von Open-Source-Software könnte für mehr Qualität bei der IT-Infrastruktur abseits der Zentren sorgen, denn sie hängt weit weniger von der räumlichen Präsenz, den Schulungen und der Unterstützung eines Herstellers ab. Die Detailkenntnisse zu bestimmten Open-Source-Produkten werden von Fachkräften typischerweise im Selbststudium und im Dialog mit einer Community erworben, die rund um den Erdball verstreut ist - besonders, wenn die Ausübenden keinen Hochschulabschluss in Informatik oder einem verwandten Feld haben. Unterstellt man, dass es auch im ländlichen Raum einen Markt für hochwertige IT-Dienstleistungen und -Fachkräfte gibt, dann eignet sich eher Open-Source-Software zur Verwendung.⁷ Es ist zu unterstellen, dass der Unterversorgung mit IT-Fachkräften in ländlichen

⁶ FdUK: „ IBM ist auch stark als Hardware-Hersteller positioniert, der hochgradig vertikal integriert ist und somit ganze „Produktpakete“, bestehend aus Hardware und Software, anbietet.“

⁷ FdUK: „ Die räumliche Entfernung von den Ballungszentren wird nicht unbedingt dazu führen, dass in ländlichen Regionen OSS verwendet wird, da OSS nicht notwendiger Weise „einfacher“ oder „wartungsfreundlicher“ ist, eher im Gegenteil (siehe auch weiter unten).“ (Anm.: bezieht sich auf benutzerfreundlichkeit)

Regionen ähnliche Strukturen zugrunde liegen wie beim Mangel an (Fach-)Ärzten.⁸

2.3.3. Altersverteilung der IT-Fachkräfte

Grundsätzlich gilt, dass die Bereitschaft zum Umstieg in andere, nicht unmittelbar verwandte Sparten mit dem Alter abnimmt. Daher sind ältere Experten oft mit älteren Technologien befasst. Da Open-Source-Technologien und Verfahren grundsätzlich ausgereifter und oft mehrere Jahrzehnte alt sind, die Standards nach bekannten und bewährten Ansätzen weiterentwickelt werden (Beispiel hypertext transfer protocol), sind die zugrunde liegenden und darauf aufbauenden Technologien langlebiger und Investitionen sicherer.⁹ Ältere Fachkräfte behalten in der Open-Source-Welt (hier eigentlich: Unix-Welt) ihre Kompetenz; im Augenblick ist die erste Kohorte dieser Experten dabei, in Pension zu gehen, hätte aber von der Qualifikation keine Probleme, am Arbeitsmarkt zu bleiben. Open-Source-Software könnte ein Aspekt bei der Beschäftigung älterer Arbeitnehmer sein.

Das Modell, ältere IT-Fachkräfte als Projektleiter und Konsulenten einzusetzen, wird in vielen Fällen funktionieren, ist allerdings nicht so universell durchführbar wie in anderen Branchen. Die Gründe dafür haben mit der Psychologie und Natur der IT-Aufgaben, den damit verbundenen Kenntnissen und Charakterzügen der Ausübenden zu tun. Die Qualitäten eines Projektleiters (führen, organisieren, kommunizieren) oder Konsulenten (Selbst-Marketing, Verhandlungsgeschick, Lagebeurteilung) fehlen vielen technisch orientierten Spezialisten; typischerweise eines der Motive für ihre Berufswahl überhaupt. Die Förderung von Open-Source-Software kann diese Situation insofern entschärfen, als dass die einzelnen Produkte und Standards den Spezialisten einen langlebigeren und sichereren Markt garantieren. Es gibt auch Hinweise darauf, dass die Akquisition von Wissen bei Open-Source Technologien mehr Motivation, Selbständigkeit und Kommunikationsfreudigkeit erfordert als die Frontal-Modelle, die für proprietäre Software üblich sind. In welchem Umfang in diesen Punkten Einflussnahme durch die öffentliche Hand wünschenswert oder machbar ist, muss gegebenenfalls weiter diskutiert werden.

⁸ FdUK: „Der Ärztemangel ist zum Teil auch auf Eintrittsbarrieren und eine restriktive Politik der entsprechenden Interessensvertretung zurückzuführen („Kassenverträge“). Zur Beseitigung fordern manche Experten eher weniger politischen Eingriff als mehr.“

⁹ FdUK: „Das ist empirisch nicht unbedingt so - da OSS oft Individualsoftware ist und überdies auch die Standards vielfach Änderungen unterworfen sind, ist auch eine Investition in OSS leider relativ risikoreich für Unternehmen - daher wird in der Regel von den Unternehmen generell zu wenig im Softwarebereich investiert.“

2.4. IT-Kostenfaktoren bei KMU

2.4.1. Hardware

Die Kosten für Hardware sind im Vergleich zu früheren Jahren praktisch vernachlässigbar und stehen auch in keinem Verhältnis zu denen für Software¹⁰ und besonders den Personalkosten für die Administration.

2.4.2. Software

Der finanzielle Aufwand für Software hängt nicht zuletzt von der Kompetenz des Einkäufers ab. Grundsätzlich gilt aber, dass die Kosten für (proprietäre) Software ständig steigen und die Lizenzen gleichzeitig restriktiver werden. Die Folgekosten für Upgrades, Ausfälle durch Virenepidemien oder Software-Fehler lassen sich weder für die Wirtschaft noch für einzelne Betroffene nur schwer absehen.

Trotz dieser durchaus bekannten und oft bemängelten Nachteile proprietärer Software schrecken die meisten KMU vor dem Einsatz von Open-Source-Software zurück. Die Gründe dafür sind in Abschnitt 4.3. näher erläutert.

2.4.3. Outsourcing

Im Augenblick noch ist „Outsourcing“ eher mit proprietären Technologien verknüpft (Unisys, Sun). IBM beispielsweise -unter anderem prominente OSS-Firma - exponiert sich als Anbieter von OSS-basierten Lösungen. Grundsätzlich wird Outsourcing, soweit es über die Inanspruchnahme trivialer Web-Dienste (Hosting, Gestaltung der Web-Site) hinausgeht, eher von großen Firmen genutzt als von KMU. Die Gründe dafür sind ähnlich denen, die unter Abschnitt 2.2. aufgeführt sind: KMU nehmen ihre IT-Infrastruktur nicht als solche wahr, sondern sehen die einzelnen Komponenten als Arbeitserleichterungen, die an die IT-unabhängigen Geschäftsprozesse angepasst werden statt umgekehrt. Da Outsourcing, wie es hier verstanden wird, die Bereitstellung von IT-Infrastruktur durch einen dem Unternehmen fremden Lieferanten bedeutet, ist dieser Sachverhalt nicht überraschend.

Outsourcing in diesem Sinne erscheint gerade für kleinere Unternehmen als idealer Weg zu adäquaten, in optimaler Weise IT-gestützten Prozessen zu sein. Ein einziger Lieferant kann mehrere, einander ähnliche Unternehmen als Kunden haben und innerhalb seiner Prozesse Know-how, Investitionen und Hardware nutzen, die ein einzelnes kleines Unternehmen nicht auslasten könnte, und als Spezialist für die einwandfreie Funktion der IT-Infrastruktur garantieren.

¹⁰ FdUK: „ Auch die Softwarekosten (Lizenzen) machen nur einen sehr geringen Anteil der „total cost of ownership“ (TCO) aus, in der Größenordnung vergleichbar mit den Hardwarekosten. Der Bereich Wartung müsste genauer spezifiziert werden, um diesen Anteil an den TCO trennscharf der Rubrik „Softwarekosten“ zuzuordnen.“

Da in so einem Modell nur das Resultat zählt, spielt die verwendete Technologie für KMU keine große Rolle, was eher Open-Source-Software begünstigt als proprietäre Software (Zuverlässigkeit, Kosten, Zukunftssicherheit). Ein KMU, dem typische Open-Source-Produkte zu „technisch“, zu „mysteriös“, erscheinen und der sie beim Einsatz durch eigene Leute im eigenen Haus nicht nutzen würde, hat vermutlich keine Einwände, äquivalente Dienste beim Outsourcing vom Lieferanten, dem solche Bedenken fremd sind, entwickeln und warten zu lassen - besonders, wenn es geringere Kosten bedeutet und die Resultate effektiver sind.

2.5. Bewertung der Ergebnisse der Software-Studie 2000 im Zusammenhang mit Open-Source-Software

2.5.1. Services, KMU und OSS

Der Einsatz von OSS in KMU ist eher die Ausnahme; selten wird dort IT optimal genutzt. Wie schon in Abschnitt 2.2 erwähnt, liegt das nicht an praktischen, sondern an gewachsenen Gegebenheiten. Die Open-Source Technologie (Linux, Apache, sendmail, etc.) baut auf der bewährten und sehr ausgereiften Unix-Systemphilosophie auf (Bell Labs, 1969), die sich seit den frühen 80ern in der Groß-EDV sehr schnell verbreitet hat. Für große Unternehmen, bis Mitte der 80er Jahre die einzigen industriellen Anwender von IT, ist der Gedanke, für die Wartung ihrer IT-Infrastruktur erhebliche personelle und finanzielle Mittel zu investieren weder neu noch erschreckend. In diese Landschaft fügen sich hochwertige OSS-Produkte mehr oder weniger natürlich ein und liefern gewohnte Leistung in gewohnter Weise zu geringeren Kosten.

Für KMU ist der Einsatz von IT um völlig andere Paradigmen aufgebaut. PCs werden vom typischen kleinen oder mittleren Unternehmen erst seit Mitte der 80er Jahre als Büromaschinen genutzt. Die Vorstellung von IT als INFRASTRUKTUR, um die herum sämtliche Geschäftsprozesse in Hinblick auf optimalen IT-Einsatz aufgebaut werden müssen, hat in kleineren und mittleren Unternehmen keine Tradition. Die von KMU eingesetzte IT wird in eher improvisierter Weise in nicht-formalisierte Abläufe mehr oder weniger „passend“ eingegliedert. Der Hauptlieferant dieser Systeme ist Microsoft, die selbst keine Geschichte als IT-Betriebsberater oder Entwickler von Gesamtkonzepten hat (wie etwa IBM oder SAP).

Diese Umstände könnten langfristig ein technologisches Abdriften der KMU zur Folge haben. So gesehen würde sich jede Verbesserung der Situation auf lange Sicht lohnen. Der Einsatz oder Umstieg auf Open-Source-Software fordert vom KMU vorübergehend hohe Investitionen in Dienstleistungen und Umstrukturierung seiner Geschäftsprozesse; die Scheu vor diesen Veränderungen ist verständlich. Informations-Maßnahmen würden hier

vermutlich sehr schnell greifen, wenn es gelingt, aufzuklären und von den Vorteilen zu überzeugen.¹¹

¹¹ FdUK: „ Oft zwingen allerdings Produkte zur betriebswirtschaftlichen Führung von Unternehmen (z.B. SAP) die betrieblichen Prozesse in ein enges Korsett - die Einführung derartige Business-Software in der (betrieblichen) Verwaltung kann bei einem etwaigen Ausstieg organisatorische Probleme verursachen, weil die betrieblichen Prozesse oft erst an die verwendete Software angepasst werden mussten. Dadurch entsteht ein gewisser „monopolistischer“ Schutz des bestehenden Produkts, weil die Umstellung mit Kosten verbunden wäre. „

3.1. Einsatzgebiete von Open-Source-Software

Grundsätzlich unterscheidet sich Open-Source-Software in den Anwendungsgebieten nicht von denen proprietärer Software, denn Open-Source ist eine Methode zur Organisation von Entwicklern und dem Vertrieb von Software. In der Praxis haben die erfolgreichsten Open-Source Projekte folgende Merkmale.

1.) Die Software ist horizontal

"Horizontal" im Sinne der Betriebswirte, die darunter breite Anwendungsgebiete (Betriebssystem, grundlegende Internet-Dienste, e-Mail, etc.) verstehen, die von einer großen Zahl von Menschen genutzt werden. Beispiele dafür: Apache Webserver, Unix, Linux, BSDs, BIND , etc.

2.) Die Software ist selbst programmierbar

Unix ist das Betriebssystem für Programmierer, hieß es einmal, und die meisten Unix-Pakete lassen sich nicht nur flexibel konfigurieren, sondern auch programmieren. Das hat in der Regel zur Folge, dass die Gründer und Urheber eines bestimmten Software-Projekts nur die rohe, nicht programmierte Software zur Verfügung stellen und die Community einzelne Module zur Verfügung stellt, die jeweils wenig Aufwand bedeuten, aber in Summe der Software ihren Wert geben (Emacs, Apache, Zope). Einige der beliebtesten Open-Source Produkte sind Programmiersprachen (CLisp, Python, Perl, Ruby), in denen von der Community umfangreiche Sammlungen von wiederverwendbaren Unterprogrammen geschrieben werden.

3.) Die Software ist modular

Unix ist ein sehr modulares Betriebssystem und ermuntert zu modularer Software. Ein individueller Entwickler kann um so leichter ein Modul entwickeln, um so besser es sich konzeptionell von anderen Modulen trennen lässt. Obwohl die technischen Details hinter dieser Unabhängigkeit sehr verschieden sein können, vereinfacht Modularität die Arbeit innerhalb der Community enorm. Beispiele: Linux' bin-utils (einzelne, völlig isolierte Programme, die über Zeichen kommunizieren), GIMP (ein Programm zur digitalen Bearbeitung von Photos, einzelne "Filter" kommunizieren mit dem Hauptprogramm und sind ansonsten voneinander unabhängig)¹²

4.) Die Software ist für technisch orientiertes Publikum¹³ interessant

¹² FdUK; „ Ein proprietäres Beispiel und Vorreiter dieser Philosophie sind u.a. Adobe-Produkte (Plug-Ins für Photoshop etc.).“

¹³ FdUK; „ Das wird auch von IT-Experten und in Befragungen bestätigt: OSS stellt oft hohe Ansprüche an die Qualifikation der User und ist oft nicht für den „einfachen“ Benutzer optimiert.“

Obwohl es zahlreiche Open-Source Produkte gibt, die von Laien verwendet werden können, welche nichts von Programmiersprachen oder Betriebssystemen verstehen, erreichen sie meistens nicht die Reife und ziehen keine umfangreiche Community an. Ausnahmen gibt es, aber die Projekte basieren meistens auf Subventionen von Firmen (AOL: AOL Server, StarOffice; Netscape: Mozilla).

Open-Source Produkte wird man in der Regel nicht finden, wo umfangreiche Spezialkenntnisse erforderlich sind, bei Produkten, die nur eine enge Nische bedienen, und die für Programmierer uninteressant sind. Beispiele: Verschnittoptimierung für Holzbretter, Kraftwerksbau, Prozessleittechnik, etc. Diese Regel ist nicht unumstößlich; jeden Tag kann eine Gruppe von Programmierern entscheiden, Software zu veröffentlichen, die alle hier genannten Kriterien erfüllt, aber unter eine Open-Source Lizenz gestellt ist. Wie schnell und ob überhaupt sich diese Software dann gegen proprietäre Konkurrenz behaupten kann, hängt von der Hingabe der Autoren bzw. von der Community ab. Der Umfang der Community selbst hängt von der Zahl der interessierten Anwender ab, die programmieren können und bereit sind, Zeit und Aufwand zu investieren. In Nischenmärkten, die Spezialkenntnisse erfordern, jedoch von Software-Entwicklung unabhängig sind, ist diese Chance sehr gering.

3.2. Metriken für Wirtschaftlichkeit und Qualitätssicherung von OS

1.) Qualität der (Open Source) Software

Prinzipiell gelten für Open-Source-Software dieselben Kriterien zur Beurteilung von Qualität wie für proprietäre Software. Im Groben lauten diese Kriterien: Ausfallsicherheit, Einbruchsicherheit, Vollständigkeit, Minimierung von Programmierfehlern, Flexibilität und Benutzerfreundlichkeit. Die Wirtschaftlichkeit von Software ist schwer zu messen, und bei Open-Source-Software nicht unumstritten, aber langsam geraten die Verfechter der Meinung „Open-Source-Software ist im Betrieb teurer als proprietäre Produkte“ in die Defensive. Bei der Bewertung von Open-Source-Software gegenüber proprietärer Software sollte man ALLE Kosten im Auge behalten; beispielsweise auch Virenepidemien, die in der Open-Source-Welt noch nie aufgetreten sind.¹⁴ Bei der Zahl der Software-Fehler sollte man nicht nur die Fehler zählen, sondern auch bewerten, welche Kosten sie verursachen, wie lange die Behebung dauert und wie viele Benutzer von dem Fehler betroffen sind. Bei der Einbruchsicherheit sollte man nicht nur die Zahl der Einbrüche ermitteln, sondern auch, welche Kosten dadurch verursacht wurden, und wie wichtig der kompromittierte Server ist.

¹⁴ FdUK: „Diese „Virensicherheit“ hängt aber derzeit vor allem mit der geringen Verbreitung oder mangelnden Interoperabilität zusammen. Eine weitergehende Verbreitung von OSS würde voraussichtlich leider auch die Virenanfälligkeit erhöhen.“

2.) Management-Techniken und Arbeitsorganisation

Die folgende Gegenüberstellung der Organisation der Arbeit in Open Source- und proprietären Software-Projekten zeigt, dass ein direkter Vergleich sehr schwierig ist (Erläuterung folgt nach der Gegenüberstellung)

2.1.) Proprietäre Software

Proprietäre Software folgt einem hierarchischen Entwicklungsmodell bei dem die Fachkräfte die Vorgaben und Kundenwünsche über Vermittlung des Managements erhalten. In vielen Fällen werden die Vorgaben von der Marketing-Abteilung entwickelt, die Kundenwünsche erraten oder Leistungsmerkmale durch Beobachtung der Presse und der Mitbewerber festzulegen. Besonders in großen Projekten sind die Software-Entwickler oft zu reinen Befehlsempfängern degradiert, die keinerlei Einfluss auf die verwendeten Programmiersprachen, Ansätze und Werkzeuge haben - was die Motivation und damit die Produktivität mindert. Der überwiegende Anteil all dieser Software-Projekte (75 Prozent) erreicht die Vorgaben nie.

Die Benutzerdokumentation kommerzieller Software ist meistens so mangelhaft, dass unzählige Fachverlage erhebliche Umsätze erzielen, indem sie diese Lücke füllen. Software die in Unternehmen für den internen Gebrauch entwickelt wird, erhält oft überhaupt keine Dokumentation. Da die Software-Entwickler unter enormen Zeitdruck stehen, ist der Code schlecht oder gar nicht dokumentiert, was die Auftraggeber von den Schlüsselfiguren eines Projekts abhängig macht - niemand sonst weiß, wie die Software funktioniert und wie man sie verbessern oder erweitern kann.

2.2. Open Source Software

Bei Open Source Software, auf der anderen Seite, ist die Motivation sehr hoch - freiwillige Programmierer stehen in Wettbewerb miteinander, und ihre Arbeit wird ununterbrochen von Gleichgesinnten bewertet und diskutiert. Der konzeptionell und handwerklich beste Code setzt sich durch. Da die Programmierer jene Leistungsmerkmale implementieren, die sich selbst von ihrer Software wünschen, wird Open Source Software nicht am Markt vorbei entwickelt. Auftretende Fehler werden behoben, sobald sie erkannt und diskutiert wurden. Sobald sich jemand findet, der eine Gelegenheit sieht, sich durch eine Verbesserung zu profilieren oder genug unter einem Mangel leidet, wird der Fehler behoben, was oft nicht länger als einige Stunden dauert. Die Dokumentation der Software wird von jenen geschrieben, die einen kurzen oder langen Beitrag über Aspekte der Software verfassen wollen, die sie am meisten schätzen oder über die sie am meisten wissen. In der Praxis werden Kataloge der einzelnen Funktionen und Leistungsmerkmale zunächst aus Fragen und Antworten zusammengestellt, die in Internet-Foren kommuniziert wurden. Didaktisch gut strukturierte Tutorials oder Lehrbücher werden von Enthusiasten verfasst, die ihre Werke zur eigenen Belehrung schreiben und entlang der Kritik ihrer Leser verbessern und polieren. Die Autoren ernten Prestige und können ihre so

dokumentierten Kenntnisse durch Schulungsverträge oder Verträge mit Verlagen verwerten.

Diese Gegenüberstellung sollte deutlich machen, dass ein Vergleich der beiden Modelle durch in der Wirtschaft übliche Verfahren vermutlich scheitern wird. Die gängigen Produktivitätsmessungen nach Zeilen Code pro Tag und Programmierer oder Fehler pro Zeile Code, etc. sind denkbar, werden aber vermutlich keine aussagekräftigen Ergebnisse liefern. Nimmt man aber als Maßstab die hohe Qualität, Zuverlässigkeit und Vollständigkeit des resultierenden Codes, so zeigt sich, dass der Open Source-Prozess dem hierarchischen organisierten Management überlegen scheint.

Diese Beobachtung funktioniert überall dort nahezu perfekt, wo umfangreiche Communities generische Produkte wie Linux oder Apache pflegen. In der Regel ist es aber so, dass man auch mit Open Source-Nischenprodukten gut arbeiten kann, wenn man bereit ist, etwas Aufwand zu investieren, intensiven Kontakt mit der (kleinen) Community pflegt und motiviert und kompetent ist. Konkurrierende proprietäre Nischenprodukte sind typischerweise ebenfalls im selben Verhältnis mangelhafter, weswegen auch auf diesem Terrain Open Source-Produkte immer beliebter werden.

Das wichtigste und schwierigste ist aber, dass beim Vergleich von Open-Source-Software mit proprietärer Software möglichst ähnliche Szenarios gegenüberstellt und präzise erfasst werden. Eine Auflage, an die sich nicht alle Studien in dieser Sache gehalten haben.

3.3. Zusammenhang von Technik und Gesellschaft im Bezug zum Open-Source-Konzept. Kreatives und produktives Potential der OS Entwicklergemeinde

Technik existiert, entgegen anderslautender Ansichten, nicht unabhängig von der Gesellschaft und ist nicht selbstgesteuert. Technik entsteht in erster Linie durch Anforderungen aus der Praxis. Die weitverbreitete Haltung, der Einzelne sei der technischen Entwicklung oder individuellen Apparaten hilflos ausgeliefert, entsteht zum einen durch mangelndes Verständnis elementarer technischer und gesellschaftlicher Zusammenhänge und durch die sprichwörtliche Unbedienbarkeit und Fehleranfälligkeit technischer Artefakte (Computerprogramme, Videorekorder, Heizungssteuerungen, etc.). Gleichzeitig erreichen viele technische Neuerungen nach einiger Zeit furchterregende Verbreitung und führen zu umwälzenden Neuerungen (Schienenverkehr, Straßenbau, Automatisierung, Antirezeptiva, Gentechnologie, etc.), auf die der Gesetzgeber reagieren muss. In vielen Fällen ist unklar, bis zu welchem Grad staatliche Regelung möglich oder wünschenswert ist; wie diese Regelungen aussehen sollen ohne den Markt zu verzerren. Mit der zunehmenden Bedeutung von Computerprogrammen hat diese Notwendigkeit auch den IT-Bereich erfasst. Open-Source-Software zeichnet sich in diesen belangen durch folgende Eigenschaften aus.

- 1.) Open-Source-Software kann zu einem erfolgreichen Vorbild für die Organisation kreativer Arbeit werden.

Im Augenblick gibt es Bestrebungen, den Open-Source-Prozess auch auf andere Informationstechnologien im weitesten Sinne anzuwenden; speziell auf Content wie Lehrbücher, Musik, 3D-Modelle und sogar Schaltkreise (www.opencores.org). Diese Versuche stecken noch in den Kinderschuhen, sehen aber vielversprechend aus. In dem Maße, wie sich Open-Source-Software durchsetzt, könnten sich andere Kreative dazu ermuntert fühlen, ihre Werke frei zu zirkulieren.¹⁵

- 2.) Die Transparenz von Open-Source-Software setzt wichtige Zeichen für die Transparenz der Prozesse in der Verwaltung.

Im Augenblick steht das erklärte Ziel der einzelnen Regierungen, transparenter und bürgernäher zu werden in krassem Gegensatz zu dem Bild, das die meisten Menschen von gesellschaftlichen Prozessen haben -- besonders, wenn es um Entscheidungen auf EU-Ebene geht. Open-Source-Software könnte helfen, im Bestreben nach mehr Transparenz und mehr Glaubwürdigkeit zu erlangen.¹⁶ Dieser Zusammenhang mag konstruiert erscheinen, aber man muss sich vor Augen halten, dass beispielsweise die Software des Finanzamts nicht einfach ein Betriebsmittel oder Werkzeug ist, sondern die Software IST GEWISSERMASSEN DAS FINANZAMT. Die Offenlegung der Software ist gleichbedeutend mit der Offenlegung der Prozesse im Finanzamt selbst. Diese Überlegung gilt für die gesamte Verwaltung.

- 3.) Open-Source-Software schafft die ideale Umgebung für Innovation¹⁷

Jeder ist eingeladen, Open-Source-Software nach eigenen Vorstellungen zu verbessern. Das bedeutet eine umfangreiche Basis für die INTEGRATION eigener Ideen. Da Innovationen selten neue Produkte oder Produktkategorien betreffen oder so radikal sind, dass sie ein ganzes Genre transzendieren, erfordern die meisten innovationen Produkte, in die sie zur Erprobung eingestuft werden können. Diese inkrementellen Fortschritte ergeben aber in Summe äußerst fortgeschrittene Produkte, die in unzähligen Details innovativ sind. Der Erfinder ohne Open-Source-Software steht vor dem Problem, seine vielleicht unscheinbare Idee als Trockenübung implementieren zu müssen, um dann einen Hersteller proprietärer Software überzeugen, der sie in sein Produkt integriert. Mit Open-Source-Software hat

¹⁵ FdUK: „Man muss aber auch bedenken, dass ein Fehlen von Urheberrechten in der Regel zu einer Unterversorgung mit den betreffenden Gütern führt (so genannte „öffentliche Güter“) und in der Regel von Ökonomen als Marktversagen eingestuft wird. OSS wird sich nur behaupten können, wenn der Entwickler von OSS parallel dazu auch gewinnbringende Produkte oder Dienstleistungen verkaufen kann.“

¹⁶ FdUK: „Derzeit sind aber viele Regierungen bestrebt, durch die Einführung von „Business“-Software ein modernes und effizientes Image der öffentlichen Verwaltung zu fördern. Mehr Transparenz und Offenheit (z.B. durch die zitierten „Open-Source-Prozesse“) wären in der Tat auch in diesem Zusammenhang sehr wünschenswert.“

¹⁷ FdUK: „Das wird allerdings in der ökonomischen Literatur vielfach bezweifelt. Der Mangel an ökonomischem Gewinnpotential führt in anderen vergleichbaren Branchen eher zu einer Reduktion der Innovationsrate. Auf der anderen Seite erleichtert die Kenntnis des Quellcodes die Verbesserung bestehender Software erheblich. Ein schwieriges wirtschaftspolitisches Problem, vielleicht dadurch zu lösen, dass Teile von Software OSS sind, andere geschützt (siehe auch Apple-Strategie).“

dieser Erfinder eine reiche Auswahl an Anwendungsgebieten und ein großes Reservoir an bestehender Software, in die er seine Idee und seine Prototypen integrieren kann. Sobald das Detail in eine bestehende Applikation eingenistet und angenommen wurde, gehört sie zum etablierten Stand der Technik und findet weite Verbreitung.

3.4. Standards und Interoperabilität von Open-Source- Software

Der Open-Source-Prozess in der Entwicklung von Standards

Seit etwa 1999 gibt es in der öffentlichen Verwaltung einen weltweiten Trend, der den Einsatz von OSS im eigenen Gebrauch von IT fördert. Die Motive hinter diesen Anstrengungen reichen von der Aussicht auf reduzierte Kosten über Zweifel an der Einbruchssicherheit und Zuverlässigkeit von proprietärer Software bis zum Wunsch nach Unabhängigkeit. Ein extremes Beispiel für derlei Bestrebungen ist die Volksrepublik China.

Sogar von Verfechtern und Kennern von freier Software im e-Government oft übersehene Aspekte sind die der Transparenz und der Nützlichkeit des Open-Source-Prozesses in der Schaffung offener Standards für die Verwaltung und die Wirtschaft. Besonders in der EU bemühen sich die einzelnen Verwaltungen um mehr Bürgernähe, Partizipation und Transparenz. Bis zu welchem Grad die Öffentlichkeit in der Lage und bereit ist, das tatsächlich zu tun, ist nicht absehbar.

Gleichzeitig wird der Open-Source Prozess für die Entwicklung offener internationaler technischer Standards immer attraktiver. Wer die mühevollen, unproduktiven und extrem kostspieligen Verhandlungen zur Schaffung der Telephonie- und Fax-Standards (CCITT) beobachtet hat und sich für die Zukunft des Länder übergreifenden e-Governments Besseres wünscht, der wird die Adaption des Open-Source Prozesses als Segen begrüßen.

Konkret bedeutet das folgendes. Bisher ist es so, dass, sobald bestimmte Technologien öffentlichen Einsatz finden oder Länder übergreifende Kooperation gewünscht werden, internationale Verhandlungen beginnen und Experten aus der Industrie zurate gezogen. Diese Verhandlungen verlaufen oft schleppend, weil jede Partei versucht, ihren eigenen Produkten, Verfahren, Patenten oder Ideen eine privilegierte Position im Standard zu verschaffen. Das Resultat ist oft ein Standard, der in einem Konglomerat aus einander überlappenden, weil konkurrierenden, Komponenten besteht und daher schwer zu implementieren ist. Eine weitere Schwäche: große Teile werden spezifiziert, ohne die genaue Praxis der Anwendung absehen zu können.

Im Gegensatz dazu wurde das Hypertext Transfer Protocol HTTP (das Web-Protokoll) zunächst von einem einzigen Mann in der Manier eines typischen Unix Applikationsprotokolls entwickelt und verbreitet. Als eine kritische Masse von Anwendern und Interessenten erreicht und die Nützlichkeit mehr oder weniger erwiesen war, begann man mit der Zirkulation der Spezifikation in einem RFCs (Request for Comment). An dieser Debatte zu technischen Standards kann sich jeder beteiligen, und eine große Anzahl von Herstellern, Forschern, Individuen und Anwendern nimmt tatsächlich teil - unter der Aufsicht und Moderation der IETF (Internet Engineering Task Force), einer gemeinnützigen NGO. Gleichzeitig werden die Vorschläge von Freiwilligen implementiert (proprietär, als Prototypen und/oder Open-Source Produkte) und erprobt. Die besten und tragfähigsten Konzepte setzen sich durch. Bei HTTP war der Übergang zum „Fertigen“ (Standards werden genaugenommen nie fertig) fließend und ständig von mehr oder weniger brauchbaren Produkten begleitet; die Erfahrungen und die Praxis flossen in die Debatte und damit in den Standard ein.

HTTP ist ein sehr breites, horizontales Produkt, da es grundlegende Infrastrukturdienste leistet. Die IETF ist nur mit solchen Spezifikationen befasst, die für das Internet von allgemeinem Interesse sind. Für e-Government wird es notwendig sein, spezielle Protokolle, Datenformate und Prozeduren zu schaffen, die nur für die öffentliche Verwaltung Bedeutung haben und dem Datenaustausch zwischen Regierungen, Behörden und Ämtern untereinander oder mit Individuen oder Firmen dienen. Diese Standards sollten nach dem Modell IETF, nicht in der traditionellen Weise geschaffen werden. Brauchbare Prototypen sollten am Anfang jeder Diskussion stehen; die Praxis sollte die Diskussion begleiten, die Anwender selbst sollen die Entwicklung der Prototypen steuern und an der Debatte teilnehmen (bei HTTP entwickelten die ersten Anwender diese Prototypen selbst, bei Verwaltungssoftware müssten die Anwender wohl durch kenntnisreiche Experten vertreten werden); die Prototypen sollten so weit wie möglich im wirklichen Tagesgeschäft, nicht in bloßen Gedankenexperimenten oder Labortests verwendet werden; wer sich berufen fühlt an der Erörterung teilzunehmen, sollte Gelegenheit dazu bekommen -- besonders, wenn er den laufenden Code vorweisen kann.

4.1. Die Zunahme von Open-Source-Software in der Wirtschaft führt zu einem wachsenden Bedarf an OS-Fachkräften

Bei der Untersuchung darüber, wie weit sich Angebot und Nachfrage nach Open-Source Spezialisten in Zukunft decken können, wie schnell sich Individuen und Unternehmen auf Open-Source-Tools und Methoden umstellen können, muss man die *Anwendung* von Open-Source-Produkten (typischerweise Server-Software wie Apache, Sendmail auf Linux oder *BSD) von der *Entwicklung* mit und für Open-Source-Produkte unterscheiden. Der OSS-Servermarkt für kleine und mittlere Unternehmen stellt eher geringe Anforderungen an einen IT-Fachmann. Kenntnisse können schnell angepasst werden, sobald sich der Arbeitgeber zu einer Migration auf Open-Source-Produkte entscheidet.

Die Wachstumsrate für Open-Source-Software ist besonders am Servermarkt sehr hoch. Engpässe bei Arbeitskräften waren hier nur während der dot.com-Bubble zu beobachten. Das liegt vermutlich am großen Reservoir professionell brachliegender Kenntnisse der zentralen Produkte Linux, Apache, Sendmail, PHP und Perl, die sich viele IT-Professionisten aus Liebhaberei aneigneten, bevor sich Gelegenheit bot, diese Kenntnisse beruflich zu verwerten. Es kommt oft vor, dass sich IT-Spezialisten im Job mit proprietären Produkten befassen, in der Freizeit aber mit Open-Source-Produkten experimentieren. Diese versteckte Know-how hat nicht immer die für die Praxis erforderliche Tiefe, kann aber schnell genug weiter entwickelt werden, sobald sich eine Gelegenheit bietet, die Produkte und Kenntnisse auch beruflich zu nutzen.

Bei der *Entwicklung* von Software ist die Situation nicht so klar wie im reinen Server-Anwendungsbereich. Zum einen sind die typischen Programmiersprachen und Ansätze der proprietären Software-Welt so weit von denen der in der Open-Source-Welt üblichen entfernt, dass die Umstellung existierender Projekte wesentlich länger dauern würde als die Umstellung einer (auch großen) Server-Landschaft. Zwar gibt es für praktisch alle Programmiersprachen der kommerziellen Windows-Welt entsprechende Open-Source-Äquivalente. Eine Abkehr von den üblichen Bibliotheken, Betriebssystemerweiterungen und anderen Ressourcen erfordert vom Unternehmer hohe Kosten und vom Individuum den Erwerb umfangreicher Spezialkenntnisse.

Im Augenblick wird proprietäre Software typischerweise in C++ oder Java entwickelt und läuft unter Microsoft Windows. Um Routineaufgaben zu erleichtern, greifen die Entwickler auf objektorientierte Bibliotheken zurück, die entweder von Microsoft oder von Drittherstellern verkauft werden (Microsoft Foundation Classes, Objective Grid, WinWidgets). Viele von diesen Werkzeugen sind ein de facto-Standard und bedienen einen horizontalen Markt. Diese Austauschbarkeit der verwendeten Produkte hat für die Industrie den Vorteil, dass die Kenntnis über diese Produkte vorausgesetzt werden kann und die Entwickler austauschbar macht.

Umgekehrt kann der Entwickler davon ausgehen, seine jahrelange Pflege dieser Kenntnisse immer wieder einbringen zu können. Der Nachteil ist, dass diese Produkte nicht so langlebig sind wie Open-Source-Produkte und nicht von einer Community, sondern in den meisten Fällen von einem einzigen Hersteller (und seinen liebsten Kunden) gepflegt werden. Dieser Markt von proprietären Werkzeugen zur Entwicklung proprietärer Software ist gut eingespielt, doch weitestgehend monopolisiert - mit allen Nachteilen die daraus erwachsen.

Im Prinzip ließe sich diese proprietäre Software mit entsprechenden Open-Source-Tools entwickeln. Für die heutigen Mainstream-Spezialisten allerdings würde das ein Verzicht auf ihre vertrauten Bibliotheken und Werkzeuge bedeuten, und eine entsprechende Einarbeitungszeit in diese neue Software. Gleichzeitig würde subjektiv der Komfort sinken, weil Open-Source-Software im Augenblick einfache Bedienbarkeit nicht als Schwerpunkt hat¹⁸. Dieser Einwand hat wenig Gewicht, denn Anwendungen werden in der Open-Source-Welt üblicherweise nicht mit diesen Werkzeugen entwickelt. Zum einen kommen andere, wesentlich flexiblere Programmiersprachen zum Einsatz, die einen großen Teil der Bibliotheken überhaupt überflüssig machen (Perl, Python, tcl), zum anderen stehen dem Entwickler gewaltige Reservoirs an hochwertigen und hoch spezialisierten Modulen zur Verfügung, für die es in der Welt der proprietären Software noch keine Entsprechung gibt. Diese Entwicklungswerkzeuge und die Open-Source-Betriebssysteme (allen voran Linux) sind um völlig andere Paradigmen herum aufgebaut, und eine Adaption erfordert vom Entwickler umfangreiche Anpassung seiner Ansätze und Kenntnisse. Dass diese Methoden und Werkzeuge nicht breiter eingesetzt werden hat keine praktischen Gründe, sondern liegt eher an dem Beharrungsvermögen der IT-Branche¹⁹ an sich. [Anmerkung: Die Situation ist noch um eine Facette reicher. Die meisten Open-Source-Programmiersprachen, und alle bedeutenden Open-Source-Programmiersprachen, laufen auch unter Microsoft Windows. Open-Source ist auch hier keine Frage des Entweder-Oder, sondern kann sehr fein variiert werden.]

¹⁸ FdUK: „Das könnte auch daran liegen, dass der Entwickler vorwiegend für sich entwickelt (um sein spezifisches Problem zu lösen) und meist technisch gut informiert ist (Kenntnis von Konsolen-Anwendungen, keine Bevorzugung einer einfachen grafischen Benutzeroberfläche). Er hat nur geringen Anreiz, für andere Benutzer eine allgemein anwendbare Oberfläche zu entwickeln.“

¹⁹ FdUK: „Eine weitere Ursache könnte darin bestehen, dass der Markt selbst komplex ist. Auch im OSS-Bereich gibt es z.B. „monopolartige“ Eintrittsbarrieren durch sehr hohe Anforderungen an die Qualifikation der Mitarbeiter in Bezug auf bestimmte Technologien.“

Bei der Weiterentwicklung von Open-Source-Produkten und dem Umgang mit den entsprechenden Open-Source-Tools ist die Diskrepanz zwischen den privaten Interessen der Individuen und der Industrie also größer als im Server-Bereich. Viele IT-Fachleute mit akademischer Ausbildung erwerben aber praxistaugliche Entwicklerkenntnisse mit Open-Source-Produkten und engagieren sich in der Community, um nach dem Berufseinstieg vorwiegend mit proprietärer Software zu tun zu haben. In anderen Worten: im Augenblick und für die absehbare Zukunft gibt es ein größeres Angebot an Open-Source-interessierten Spezialisten als Arbeitsplätze.²⁰ Wie weit die Qualität der Kenntnisse den Anforderungen einer Businesswelt entsprechen kann, die sich zunehmend auf Open-Source-Produkte zugreift, müsste erst erhoben werden.

Was beiden Bereichen - der Entwicklung von Software mit Open-Source-Produkten und der bloßen Anwendung von Open-Source Server-Produkten - fehlt, ist die systematische, praxisnahe Ausbildung und die Zertifizierung dieser Kenntnisse. Rund um proprietäre Produkte gibt es eine von den Herstellern betriebene regelrechte Zertifizierungsindustrie, die fachspezifisches und produktspezifisches Wissen sehr eng an Einsatzmöglichkeiten und "Best Practices" koppelt und den Erwerb der Kenntnisse durch die Teilnehmer verbrieft (Novell Certified Professional, Cisco ...).

Diese Möglichkeiten fehlen sowohl den IT-Fachkräften der Open-Source-Welt als auch ihren potentiellen Arbeitgebern. Aus der Sicht der OS-Community wäre eine von der öffentlichen Hand geförderte Zertifizierung der Qualifikationen anzustreben, bis sich ein eigenständiges, von bestimmten Herstellern unabhängiges Schulungs- und Zertifizierungssystem herausgebildet hat.²¹

4.2. Stetige Diversifizierung und Spezialisierung führt zum Entstehen neuer Berufe und Berufszweige

Ein anderer Aspekt - und hier sind IT-Berufe ausnahmsweise unter anderen Branchen hervorgehoben - ist die rasche und feingliedrige Diversifizierung und Spezialisierung der einzelnen Berufsbilder. Beispielsweise war die Administration von Datenbank-, Web- und Mail-Server üblicherweise Aufgabe eines allgemein eingesetzten „Unix-Administrators“. Heute sind die einzelnen Genres jeweils eigene Jobs mit sehr verschiedenen Anforderungen. Dieser Erscheinung sollte durch ebenso spezialisierte Ausbildungen entsprochen werden, die an allen Arten von Bildungsinstituten

²⁰ FdUK: „ Das ist eine Prognose, die viele Ökonomen teilen.“

²¹ FdUK: „Initiativen wie z.B: OSS-ECDL (Europäischer Computerführerschein) klingen sehr interessant und werden von einigen Experten befürwortet. Es ist allerdings fraglich, ob tiefer greifende Zertifizierungen für Arbeitgeber tatsächlich interessant wären - die Zertifizierungen würden jedoch den OSS-Anbietern eine „proprietäre“ Einnahmequelle eröffnen, die das Marktversagen in gewisser Weise reduzieren würde. Andererseits stellt sich die Frage, ob OSS so wenig dynamisch (dafür aber rückwärtskompatibel) ist, dass die Weiterentwicklung von OSS durch derartige Zertifizierungen abgebildet werden kann.“

angeboten werden (FH, HTL, Universitäten, Erwachsenenbildung). Dazu gehört auch ein entsprechendes Zertifizierungssystem.

Vom „lebenslangen und berufsbegleitenden Lernen“ wird schon lange gesprochen, und tatsächlich ist diese Notwendigkeit für viele Berufe älter als dieser Begriff und die Voraussetzung für Karriere überhaupt. Erst jetzt zeichnet sich deutlich ab, dass unentwegtes Dazulernen und Umlernen eine für fast alle Erwerbstätigen typische Anforderung ist.²² Obwohl IT-Berufe wegen der spektakulären Fortschritte auf diesem Gebiet das Beispiel für eine schnelllebige Arbeitswelt sind, beschränken die Aussagen dieses Abschnitts ihre Gültigkeit weder auf die IT-Branche noch auf Open-Source-Software.

Ein Teil der Probleme des Arbeitsmarktes hat seine Wurzeln im starren kulturellen Verständnis der Arbeitswelt. Der geradlinige Ablauf Ausbildung-Beruf-Pension wird immer mehr durch „Patch-work“-Karrieren verdrängt. Unregelmäßige Arbeitsphasen und unterschiedliche Formen (angestellt, freie Mitarbeit, Selbständigkeit, arbeitslos, Urlaub, Freizeit, Weiterbildung, time-out) der Beschäftigung entsprechen - scheinbar - den heutigen Anforderungen. Die Arbeitswelt erfordert ein Maß an Selbständigkeit, Aggressivität, Kreativität und Kommunikation.

Entsprechend ändern sich die Anforderungen an das Bildungssystem; die Bildungsinstitute (auch die der Erwachsenenbildung und besonders die Dienstleister des AMS) sollten forciert selbständiges und kommunikatives Lernen fördern.

4.3. Das Anforderungsprofil an OS-Fachkräfte (*Kreativität, Motivation, Teamwork, Mediator*)

Gegenwärtig unterscheidet sich die Arbeit mit Open-Source-Produkten und -Ansätzen an vielen Stellen von der mit proprietären Produkten. Wie groß diese Unterschiede sind, hängt vom Einsatzgebiet und vom Ehrgeiz der Ausübenden ab. Die Unterschiede sind bedeutender, wenn es nicht um den bloßen Einsatz von Server-Produkten (Apache, Sendmail, MySQL, etc.) geht, sondern um *Software-Entwicklung*. Die Unterschiede bringen jeweils verschiedene Anforderungen an die IT-Fachkräfte mit sich.

Im Server-Bereich, auf der einen Seite, empfinden gut ausgebildete Fachkräfte die Arbeit mit Open-Source-Software oft als angenehmer und befriedigender als die mit äquivalenten (oder eben nicht ganz äquivalenten) proprietären Produkten.

Die beliebteste freie Software - Linux, Apache, MySQL, Perl, php - zeichnet sich durch eine umfangreiche und sehr produktive Gemeinde aus, durch lückenlose, immer aktuelle Dokumentation, durch detaillierte Tutorials, Fachliteratur und rund um die Uhr verfügbare Hilfestellung und Gedankenaustausch mit Gleichgesinnten.

²² FdUK: „Die Notwendigkeit permanenter Weiterqualifikation relativiert ebenfalls die Sinnhaftigkeit sehr tief greifender Zertifizierungen.“

Die entsprechenden proprietären Produkte haben oft den Ruf, für typische Anforderungen einfacher zu bedienen zu sein, aber den Fachmann vor unlösbare Probleme zu stellen, wenn es um Aufgaben abseits der Routine geht. Entfernt man sich vom lebhaften und gut gepflegten Server-Terrain, so zeigt sich, dass die Arbeit mit Open-Source-Produkten in der Regel mehr Findigkeit, mehr profunde Kenntnisse, mehr Kommunikation und mehr Motivation erfordert als die mit proprietären Produkten. Tatsächlich sind viele proprietäre Produkte in Hinblick auf Benutzerführung, Point-And-Click und Komfort optimiert.²³ Dieser Komfort geht in der Regel auf Kosten der Flexibilität und Mächtigkeit dieser Werkzeuge und kann bei komplexen oder sehr technischen Anwendungen schnell unzureichend werden. Kritiker sehen den Mangel an konzeptioneller Schlichtheit und das Fehlen ausgefeilter Benutzerführung oft als Mangel von Open-Source-Software überhaupt; sie verweisen auf den Umstand, dass proprietäre Produkte "billigeres" Personal und damit geringere Kosten ermöglicht.

Diese Form der Kostenreduktion funktioniert aber nur bei verhältnismäßig einfachen Aufgaben, die nicht über jene hinausgehen, die der Hersteller als typisch oder gängige Nachfrage definiert hat.

Die Stärken von Open-Source-Software zeigen sich dann, wenn kompetente Fachleute in der Lage sind, mit der Community in Verbindung zu treten (möglichst in englischer Sprache) um technische Belange zu diskutieren. Ein kompetenter Open-Source-Fachmann wird praktisch jedes technische Problem mit seiner Software und seine Aufgabe lösen können, wenn er in der Lage ist, es in schriftlicher Form für andere Fachleute verständlich zu formulieren und an einer Diskussion teilzunehmen, die zur Lösung des Problems führt. Je größer die Community ist, umso mehr Leute sich bereits mit ähnlichen Aufgaben befasst haben, umso wahrscheinlicher ist eine rasche Abwicklung dieses Dialogs. Dieser Ansatz erfordert natürlich ein hohes Maß an Motivation, Kommunikationsfreudigkeit, Erfindungsgeist und Kompetenz.

Führen diese Anforderungen mit steigender Popularität von Open-Source-Software zu einem Mangel an geeigneten Fachkräften?

Um diese Frage zu beantworten, darf man sich nicht vom gegenwärtigen Stand der Dinge ablenken lassen. Open-Source-Software entstand unter anderem dadurch, dass sich einzelne Entwickler Software wünschten, die es am Markt nicht gab (oder nicht zu dem Preis, den sie aufbringen konnten oder wollten). Die involvierten Techniker schufen zweckorientierte Communities um ihre Produkte, da sie den Gewinn weniger in der kommerziellen Verwertung ihres Codes sahen, sondern im Code selbst, der durch die Community gepflegt wird. Aus dieser Tradition heraus wachsen die heutigen Open-Source-Produkte weiter und ziehen Gleichgesinnte an, die durch die Arbeit in der Community „akkulturisiert“ werden (vgl. Eric S.

²³ FdUK: „Das könnte ein gutes Argument dafür sein, warum OSS (abgesehen vom reinen Betriebssystem-Bereich oder reinen Server-Bereich) vielfach als Individualsoftware vorkommt.“

Raymond, „Die Kathedrale und der Basar“). Diese Beobachtung und diese Abwicklung des Open-Source-„Geschäfts“ muss nicht für alle Zeiten für alle Teilnehmer gültig bleiben.

4.4. Fachkräfte und Telearbeit, neue Selbständige, Micro-Enterprises, Competence-Networks

Aus der IT-Branche im Allgemeinen und dem Open-Source-Bereich im Besonderen kommen Stimmen, die meinen, dass die gesetzlichen Rahmenbedingungen den nachgefragten Beschäftigungsmodellen im Weg stehen würden.

Dies betrifft die sensiblen Themen wie Arbeitszeit, Urlaubsanspruch, Bildungskarenz, Arbeitsstätte und Arbeitsort, um nur die wichtigsten herauszugreifen. Künftig fördert die Globalisierung verstärkt einen Arbeitsmarkt ohne Grenzen, der IT-Bereich ist dabei in einer Vorreiterrolle.

Eine Mehrheit der hervorragend qualifizierten und eng spezialisierten "Knowledge-Worker" werden einem Modell eines künstlerischen freien Berufs folgen. Merkmale für dieses Modell sind eine relativ teure Arbeitskraft und unregelmäßige Intervalle zwischen Akquisition, Beschäftigung, Weiterbildung und Freizeit. Diese Art zu Arbeiten kann sehr lukrativ sein und viele der Ausübenden empfinden dies individuell eher ein Mehr als Weniger an Lebensqualität. Auf der anderen Seite müssen die entsprechenden Rahmenbedingungen neu definiert und angepasst werden, das Risiko von Krankheit, Arbeitslosigkeit und Altersversorgung sollte nicht dem Individuum unterbunden sondern solidarisch organisiert sein, eine Herausforderung an die Politik.

Ein interessanter Vergleich betreffend der Organisation der Arbeit bietet sich zwischen Künstlern und IT-Fachkräften an, so könnte man sich die IT-Branche für die Zukunft ähnlich organisiert wie die kalifornische Spielfilmindustrie ("Hollywood") vorstellen. Sehr eng spezialisierte Dienstleister/Arbeitskräfte arbeiten nicht permanent für einen einzelnen Arbeitgeber, sondern ihre nachgefragten Fähigkeiten werden überall dort eingesetzt, wo sie gerade gebraucht werden.

Sowohl in der Filmbranche als auch im Open-Source-Bereich kommt es häufig vor, dass sich einzelne Unternehmen vorübergehend zu Arbeitsgemeinschaften zusammenschließen und für die Dauer eines Projektes sehr eng kooperieren - wie das bei heutigen Open-Source-Projekten auch der Fall ist. Ein weiteres Merkmal dieses "Modell Hollywoods" sind winzige Firmen, die ihre Verbindungen, Ressourcen und Know-How so geschickt und flexibel einsetzen können, dass sie gewaltige Umsätze praktisch ohne Fixpersonal, ausgedehnten Firmensitz oder Kapital erzielen. Diese "Produktionsfirmen" entsprechen vage dem oft missbrauchten und oft missverstandenen Begriffes des "Micro-Enterprise". Diese enge, oft spontane eingerichtete Kooperation basiert auf unzähligen informellen Kontakten, auf Networking, sogar zwischen konkurrierenden Unternehmen. Diese "Competence Networks" werden v.a. von Zukunftsforschern als eine neue

Form der Arbeitsorganisation und Unternehmenskooperation identifiziert, "just-in-time-cooperation".

Das "Modell Hollywood" wird hier so ausführlich strapaziert, weil es dem Open-Source-Prozess, sehr ähnlich ist und - im Gegensatz zur heutigen Open-Source-Software-Produktion - in der Filmwirtschaft bereits seit langem erfolgreich praktiziert wird.

Dazu sind entsprechende Rahmenbedingungen notwendig, vor allem auch der Bildungssektor ist gefordert. Improvisationstalent, soziale Kompetenz und "Networking" im Sinne von Selbst-Marketing und Aufbau von Kontakten muss gefördert werden.

Was der Vergleich mit Hollywood aber nicht zeigt, ist die Aussicht auf eine positive Ausformung der Globalisierung des IT-Arbeitsmarktes, von dem der Standort Österreich dank des hohen Bildungsniveaus eher profitieren könnte. Obwohl sich die Verheißungen des "Tele-Workings" für die meisten klassischen Jobs noch nicht erfüllt haben, könnten hochqualifizierte, in lockeren Verbänden arbeitende heimische IT-Fachkräfte zu einem Exportschlager werden, ohne dazu den Platz hinter ihren Tastaturen zu verlassen.

Dies skizziert, entlang welcher Linien sich öffentliche Institutionen engagieren sollten, um die Dynamik der Wirtschaft und damit des Arbeitsmarktes zu fördern. Diese Maßnahmen könnten über den Open-Source-Bereich bzw. die IT-Branche hinausgehen; Open-Source als Katalysator der Dynamisierung der Arbeitswelt.

Abseits der Diskussion um Kostenersparnis und Sicherheit - um nur zwei zentrale Elemente herauszustreichen - gibt Open-Source Anstoß zur Prozess-orientierten Arbeitsorganisation, zu entsprechenden Managementmethoden mit dem Ziel Eigenständigkeit, Engagement und Kreativität zur forcieren. Dies sind zentrale Anforderungen an eine erfolgreiche Wirtschaft und damit einen dynamischen Arbeitsmarkt in der Wissensgesellschaft.

4.5. Liste der am häufigsten eingesetzten OS-Software im wirtschaftlichen Umfeld

Die folgende Liste deckt die populärsten und reifsten Open-Source-Produkte ab; die Zahl der Anwender geht jeweils in die Millionen, im Fall von Linux, Apache, Perl und PHP sogar in die Zigmillionen. Die Communities sind (außer im Fall von Zope) so umfangreich, dass die Produkte in vielen Sprachen der Erde dokumentiert sind. Das Angebot für entsprechende Fachliteratur ist praktisch unüberschaubar. Alle Produkte zusammen (außer Zope und Python) konstituieren die weltweite Internet-Infrastruktur und zeigen sich gegen die Konkurrenz von proprietärer Software weitestgehend resistent.

4.5.1. Das Betriebssystem Linux

Linux ist ein Open-Source-Betriebssystem und wurde zum prominentesten Beispiel für die Durchschlagskräftigkeit des Open-Source-Prozesses. Linux ist eine verhältnismäßig neue Software.

4.5.2. Die *BSD Familie

NetBSD, OpenBSD und FreeBSD sind - wie Linux - einander sehr ähnliche Betriebssysteme mit gemeinsamen Vorfahren (Berkley Unix). Das Schisma, die Spaltung in Net- Open- und FreeBSD, erfolgte 1993 und wurde notwendig, weil die verschiedenen Schwerpunkte (Kompaktheit, Security, bzw. konzeptionelle Schlichtheit) sich nicht mehr in derselben Codebasis vereinigen ließen. Obwohl jeder einzelne Zweig jünger ist als Linux, ist Berkley Unix eines der ältesten Unix-Implementationen überhaupt und hat noch mehr proprietäre Nachkommen (darunter die Unices von Sun Microsystems, das Fundament für Apples OS X und das Betriebssystem der Cisco-Router). Die *BSDs sind bei weitem nicht so beliebt wie Linux, werden aber vorwiegend dort eingesetzt, wo erhöhte Zuverlässigkeit, spezielle Treiber oder einfach nur Kompaktheit gefordert sind.

4.5.3. GNU Software

Älter als Linux ist Richard Stallmans GNU Software vom Gründervater der Open-Source-Bewegung. Sie umfasst sämtliche klassische Unix-Tools (grep, awk, mail, sed, cc, etc.) und ist heute Teil von Linux. Tatsächlich hatte Stallmans Community 1991 bereits sämtliche Software geschaffen, die für ein vollständiges Unix notwendig ist -- bis auf den Kernel, den dann Linus Torvalds Community stiftete. Heute gibt es auch einen GNU-Kernel namens HURD, der ein eher verstecktes Dasein fristet.

4.5.4. Der Serverdienst Sendmail

Sendmail ist der heute mit Abstand beliebteste Mail-Server und ein sehr reifes, jedoch mit Sicherheitsproblemen behaftetes Produkt. Sendmail ist

beinahe so alt wie e-Mail überhaupt und entwickelte sich seit 1982 entlang der stetig wechselnden Anforderungen an elektronische Post.

4.5.5. Der Emacs Editor

Emacs ist ein programmierbarer Text-Editor, dessen Popularität nur vom minimalistischen (und wesentlich älteren) Konkurrenten *vi* (Teil der klassischen Unix Tool-Suite) übertroffen wird. Emacs war das erste Programm, das von Richard Stallman als Software entwickelt wurde, und das erste Programm, das überhaupt unter seine "General Public License" gestellt wurde.

4.5.6. Der Vim Editor

Vim, hervorgegangen aus dem editor *vi* (entwickelt von Berkely für IBM), ist einer der weitest verbreiteten und mächtigsten Editoren. Es gibt ein graphisches Frontend; er wird jedoch meist ohne benutzt.

4.5.7. Der Apache Webserver

Apache ist der heute am weitesten verbreitete Web-Server und wird von einer großen Community weiterentwickelt und gepflegt.

4.5.8. Die Datenbanksysteme PostgreSQL, MySQL

Beide Produkte sind relationale Datenbank Managementsysteme (RDBMS) und werden von jeweils sehr umfangreichen Communities gepflegt. MySQL ist wegen der einfacheren Bedienbarkeit aber beliebter und wird sehr oft in DB-gestützten Websites eingesetzt. PostgreSQL gilt als die Enterprise-tauglichere Software und als ernstzunehmender Konkurrent zu dem proprietären Produkt von Oracle.

4.5.9. Die Skriptingsprache Perl

Perl ist eine interpretierte Programmiersprache und mit Abstand die beliebteste Skriptsprache. Perl verfügt über eine unüberschaubare Zahl von Modulen für alle nur erdenklichen Aufgaben für Web-Server, Datenbank-Server, Mail-Server, Suchmaschinen, Spam-Filter, etc. etc. Die Infrastruktur und Community-Support für die Entwicklung, Test und Verfeinerung von Modulen sind ohne Beispiel in der Software-Industrie. Perl ist heute integraler Bestandteil des Web-Servers Apache, der sich in Perl programmieren lässt (aber nicht in Perl geschrieben ist).

4.5.10. Die Webprogrammiersprache PHP

PHP ist eine interpretierte Programmiersprache, die sich in Web-Pages einbetten lässt und so dynamische Inhalte und komplexe User-Interaktion mit Web-Sites gestattet.

4.5.11. Die Skripting Umgebung Python

Python ist eine interpretierte Scriptsprache und ein unmittelbarer (aber weit abgeschlagener) Konkurrent von Perl. Python ist jünger als Perl und stellt an den Programmierer strengere Anforderungen bei der Formulierung von Code als Perl, was diese Sprache für umfangreiche Programme und große Projekte attraktiver macht als Perl. Umfangreiche Bibliotheken und Module gibt es für alle nur erdenklichen Aufgaben, die Zahl ist aber nicht so groß wie die an Perl-Modulen.

4.5.12. Samba

Samba ist eine Neuimplementierung des File- und Print-Servers SMB von Microsoft. Samba war ein richtungsweisendes Open-Source-Produkt, weil es einen proprietären Server exakt nachbildet und gegenüber den entsprechenden proprietären Clients als vollwertiger Ersatz auftritt. Samba unterminierte Microsofts Versuch, Nicht-MS-Clients aus dem Netzwerk auszuschließen und war ausschlaggebend bei der Integration von Windows-basierten Systemen in heterogene Unix/Linux-Netzwerke.²⁴

4.5.13. Der Applikationsserver Zope

Zope ist ein Web-Application Framework und in Python geschrieben. Die Community ist vergleichsweise klein. Da Zope aber die Schaffung sehr komplexer und umfangreicher Websites und Web-Anbindungen an alle nur erdenklichen Server (auch Legacy-Systeme) gestattet, ist Zope sehr beliebt bei Spezialisten, die ihre Aufgaben mit typischen Tools und Programmiersprachen (Perl, PHP) nicht lösen könnten, Java nicht beherrschen oder nicht mischen wollen.

4.5.14. Der Linux-Desktop KDE

KDE (K Desktop Environment) ist mit GNOME einer der beliebtesten und größten Desktop Environments für Linux und andere Plattformen. Wie auch bei GNOME handelt es sich hier um einen kompletten Desktop nicht um einen Windows Manager. Beide Produkte brauchen das X Windows System.

4.5.15. Der Linux-Desktop GNOME

GNOME ist ein mit KDE kompatibles Produkt (siehe oben, KDE), das aber andere Zielsetzungen bei der Gestaltung der Benutzerumgebung hat als KDE.

²⁴ FdUK: „Samba ist ein interessantes Beispiel für die Nachahmung proprietärer Technologie durch OSS.“

4.5.16. Das UNIX Windowingsystem Xfree86²⁵

Xfree86 ist eine Open-Source-Implementation des X Window System, der klassischen Unix Thin-Client-Technologie. Xfree86 liefert zahllose Treiber für Graphikkarten und implementiert den gesamten X Window-API.

4.5.17. Der Serverdienst BIND

BIND ist der am weitesten verbreitete Domain Name Server (DNS). 80% des weltweiten DNS Resolving verlassen sich auf Technologie, die zuerst für BIND entwickelt wurde.

4.5.18 Der Serverdienst INN

INN stellt in 80% aller Fälle die weltgrößten NNTP-basierten News Groups.

4.5.19. Das Textsystem TeTeX

TeX ist die größte und bekannteste TeX Distribution für Unix. LaTeX und andere Tools verlassen sich auf dieses Produkt.

4.5.20. Das 3D Rendering Package Persistence of Vision

"pov" ist ein Rendering-Programm zur Erstellung photorealistischer Computer-Graphik. Persistence of Vision entstand zunächst als Universitätsprojekt, setzte sich aber nach einigen Jahren (und im Windschatten billiger und leistungsfähiger PCs) als Standard durch.

4.5.21. OpenOffice (vormals StarOffice)

Open Office ist das Open-Source-Äquivalent zu Microsoft Office und wird von Sun Microsystems betreut. OpenOffice beinhaltet eine Textverarbeitung (OpenWriter), ein Tabellenkalkulationsprogramm (OpenCalc), und Präsentationssoftware (Impress). Das Feature Set entspricht ungefähr dem von Microsoft Office, und Sun Microsystems legt großen Wert auf ein Look&Feel, das dem von Microsoft Office möglichst ähnlich ist. Der Import von Microsoft Office-Dateien ist möglich, wird aber meistens erst mit einer gewissen Verzögerung unterstützt, weil die Entwickler bei Änderungen die Formate erst entziffern („knacken“, „reverse engineering“) müssen. Microsoft liefert keine Dokumentation über diese Formate. Open Office gilt, anders als KOffice (siehe unten), als vollwertiger Ersatz für Microsoft Office.

²⁵ FdUK: „X Windows ist ein interessantes Beispiel für das Aufgreifen und Kopieren von Features konkurrierender Systeme: obwohl alle Beteiligten das in der Regel kategorisch abstreiten, sind die aktuellen LINUX-Desktops visuell stark an das Design der Benutzeroberfläche älterer (oder auch aktueller) Windows-Versionen angelehnt, ältere Windows-Versionen wiederum wiesen starke Parallelen zu noch früheren MacOS-Versionen auf... Dieses Kopieren von Features ist zwar nicht sehr innovativ, zeigt aber, wie bestehende Technologien genutzt oder sogar verbessert werden können, auch wenn proprietäre Produkte als geistige Quelle dienen.“

4.5.22. Abiword

Abiword ist ein MS Word-ähnlicher Textprozessor; das native Dokumentformat basiert aber auf XML. Das Feature Set ist bewußt geringer als das von Microsoft Word, Import und Export von MS Word-Dateien ist mit einigen Einschränkungen möglich.

4.5.23. KOffice

KOffice ist ein Open-Source-Äquivalent zu Microsoft Office und wurde als KDE-Applikation geschaffen. KOffice beinhaltet eine Textverarbeitung (KWord), ein Tabellenkalkulationprogramm (KSpread) und Präsentationssoftware (KPresenter). Das Feature Set entspricht ungefähr dem von Microsoft Office, ist aber noch nicht so vollständig implementiert wie das von OpenOffice (siehe oben).

4.5.24. Mozilla

Mozilla ist der beliebteste Open-Source-Internet Browser und bietet ein umfangreiches Feature Set, inklusive der Unterstützung von XLST-Stylesheets. Mozilla enthält auch einen Mail-Klienten und News Reader (für USENET).

5.1. Die Ausbildung von OS-Fachkräften und „Lebenslanges Lernen“

Die IT-Berufe sind zum typischen Beispiel für die moderne Arbeitswelt geworden, in der die Ausübenden mit der technischen Entwicklung Schritt halten müssen (vielfach sogar als „Tretmühle“ bezeichnet). In den Medien werden die Anforderung oft übertrieben, aber die Gefahr, sich relativ plötzlich in einem schrumpfenden Markt für die eigenen Kenntnisse wiederzufinden, besteht immer und kann auch für erfahrene Spezialisten zum Problem werden, wenn sie einen solchen Trend nicht rechtzeitig erkennen.

Das Schlagwort vom lebenslangen Lernen ist kein leeres, besonders, wenn sich IT-Fachkräfte in Richtung lukrativerer oder anspruchsvollerer Fachgebiete weiterentwickeln wollen (z.B. vom Web-Administrator zum Web-Programmierer). Im Augenblick unterscheidet sich der typische Werdegang eines Open-Source-Experten erheblich von dem eines Experten für proprietäre Produkte. Open-Source-Experten kommen in der Regel von einem IT-akademischen Hintergrund (Universität) und/oder haben ihre Kenntnisse zu einem großen Teil autodidaktisch erworben; Experten für proprietäre Produkte stammen meistens von einer Fachhochschule oder HTL und bilden sich in kommerziellen Kursen fort, die vom Arbeitgeber bezahlt werden.

Diese Unterschiede haben aber mit der Natur der Produkte nichts zu tun, sondern haben historische, traditionelle Gründe (siehe Abschnitt 5.3). Open-Source-Produkte zeichnen sich aber dadurch aus, dass sie langlebiger sind, und die Standards sorgfältiger herausgearbeitet werden, was individuelle Investitionen in bestimmte Kenntnisse und Spezialisierung zukunftssicherer²⁶ und attraktiver macht.

5.2. Diversifizierung der Ausbildungsmöglichkeiten

Die IT-Berufe zeichnen sich durch zunehmende Auffächerung und Spezialisierung aus, die immer feingliedrigere Nischen erzeugt. Mit der zunehmenden Komplexität und Mächtigkeit einzelner Produkte und Technologien kann ein bestimmtes Fachgebiet („Internet-Administrator“) sehr schnell unüberschaubar werden und erfordert dann Spezialisierung (in „Web-Administrator“, „Mail-Administrator“, „Intranet-Administrator“, etc.) Dieser Explosion an Nischen sollte durch eine entsprechend fokussierte Ausbildung entsprochen werden. Daneben sollte es feinere Differenzierung in

²⁶ FdUK: „Das ist in der Literatur umstritten und wahrscheinlich letztendlich stark von der spezifischen Anwendung abhängig.“

der Qualifikation geben, d.h., je nach Neigung sollte es dem Ausübenden freistehen, kürzere, billigere und spezialisiertere Curricula zu wählen („Mail-Administrator“) oder längere, allgemeinere („e-Commerce-Techniker“), bis zum vollen Universitätsabschluss.

5.3. Optionen eines restrukturierten Angebotes der etablierten Bildungsträger: Universitäten, Fachhochschulen, Höhere Schulen, Berufsschulen, Erwachsenenbildung

Die Ausbildung auf akademischen Niveau bzw. mit hohem Anteil an Selbstorganisation, komplexen Denken und analytischen Komponenten erscheint beim Arbeiten an Open-Source-Produkten von Vorteil. Tatsächlich haben die heute prominentesten Open-Source-Standards, -Betriebssysteme und -Philosophien ihre Wurzeln in Forschung und Lehre. Gleichzeitig werden Informatiker als Generalisten ausgebildet, die möglichst die Konzepte, Ideen und mathematischen Grundlagen hinter Software überhaupt verstehen sollen.

Dem gegenüber stehen praxisorientierte Ausbildungen, die mehr auf die unmittelbare Verwertbarkeit abzielen. Dort wird aufgaben- und produktorientierter operiert als in der Ausbildung von Informatikern, denen ihre Kenntnisse selten in Hinblick auf bestimmte Produkte oder praktischen, konkreten Szenarios vermittelt werden.

Umgekehrt wird in der aufgaben- und produktorientierten Vermittlung von IT-Kenntnissen in der Geschäftswelt der theoretische Unterbau konkreter Produkte und Szenarios meistens nur so weit erörtert, wie das zum Verständnis und unmittelbare Anforderungen der Materie beiträgt.²⁷

Wie gut diese Situation den Anforderungen der Wirtschaft entspricht, lässt sich schwer abschätzen und ist umstritten; wie sie verbessert werden kann, ist eine bildungspolitische Frage, die vermutlich nie für alle Kritiker befriedigend entschieden werden kann. Es gibt (nicht nur in der IT-Branche) Projektleiter und Unternehmer, die die Methoden und Resultate der Universitäten als "zu esoterisch, zu praxisfremd, zu theoretisch" bemängeln, und es gibt andere, die die Ausbildung der Fachhochschulen, der Erwachsenenbildung und der Firmenschulungen für "zu oberflächlich" halten und "Generalisten" fordern. Diese Diskussion hat mit dem Einsatz von Open-Source-Software insofern zu tun, als dass die Generalisten mit profunden Kenntnissen typischerweise als Informatiker von der Universität kommen und dort traditionellerweise eine Kultur vorfinden, die zum Einsatz von Open-Source-Software ermuntert.

²⁷ FdUK: „Das ist möglicherweise ein systemimmanentes Problem: Das AMS z.B.: bildet (aus berechtigten ökonomischen und arbeitsmarktpolitischen Erwägungen) meist keine „Fachprofis“ aus, obwohl gerade im IT-Bereich eine breitere Wissensbasis oft einen wichtigen Bestandteil der Qualifikation darstellt („Generalisten“).

5.4. Zertifizierungssysteme für OSS

Klar ist, dass Open-Source-Produkte eine aufgaben- produkt- und lösungsorientierte Zertifizierungssystematik gut gebrauchen könnten, die nach Vorbild der Herstellerkurse glaubwürdige und in der Wirtschaft anerkannte Zeugnisse ausstellt (wie die von Novell, Cisco und Microsoft). Klar ist auch, dass die Verbreitung von Open-Source-Software gefördert wird, wenn zu ihrem Einsatz in praxisnahen Bildungsanstalten ermuntert wird.

5.5. Aus der Praxis: Beispiele für neue Berufsbezeichnungen

5.5.1. LAMP-Experte

/LAMP/ steht für Linux, Apache, MySQL und PHP/Perl. Diese vier Open-Source-Produkte reichen aus, um große und komplexe Websites wie Web-basierte Produktkataloge, Content Management Systeme, Job-Börsen oder e-Commerce-Sites zu realisieren. LAMP-Experten kennen alle vier Produkte zumindest so weit, wie sie zur Verwaltung solcher Sites notwendig ist; es gibt aber zahlreiche Individuen, die kleinere Websites alleine oder große Websites in Zusammenarbeit mit anderen LAMP-Experten entwickeln können. Bei größeren Websites gibt es Arbeitsteilung und Spezialisierung auf die einzelnen Produkte; da sie aber nicht isoliert funktionieren können, muss jeder Spezialist auch Kenntnisse über alle anderen Produkte haben. LAMP ist das sehr typische IT-Tagesgeschäft hinter vielen Websites.

5.5.2. Zopist

Das Web Application Framework Zope deckt in etwa dasselbe Terrain wie LAMP (siehe Punkt 5.5.1) ab, verwendet aber völlig andere Ansätze und ist daher für eine andere Sorte von Entwicklern interessant (zwischen LAMP-Anhängern und Zopisten gibt es häufig emotionale Auseinandersetzungen darüber, welche Software die bessere ist).

5.5.3. Markup-Specialist/XML-Designer

Der Job-Title "Markup Specialist" ist heute mehr oder weniger in Vergessenheit geraten und stammt noch aus der vor XML (SGML). Besonders prominent war er nie. Zusammen mit XML, einem Standard zur Formatierung von Daten, ist die Entwicklung von XML-Produkten (d.h. konkreten Schablonen für Daten) zu einem wichtigen Beruf geworden, der viel Erfahrung und Fachkenntnis erfordert. Da XML ein offener Standard ist, sind die verwendeten Werkzeuge (Xalan, RXP, SAX, etc.) in der Regel Open-Source-Produkte.

5.5.4. CMS-Entwickler

Obwohl gerade im Bereich Content Management Systeme viele proprietäre Produkte existieren, werden sie allmählich durch Open-Source Produkte (es gibt eine große Zahl) zurückgedrängt -- besonders, wenn es um Web-basierte Content Management Systeme geht. Die beliebtesten einschlägigen Open-Source-Produkte sind PHP Nuke, Postnuke und das Zope-basierte CMF (Content Management Framework). Diese Pakete funktionieren mehr oder weniger out-of-the-box; da aber in der Praxis Anbindungen an Legacy-Systeme oder Integration mit bestehenden Redaktionssystemen notwendig ist, sind bei jeder Anwendung auch Entwickler wenigstens vorübergehend im Spiel.

5.5.5. Community Liasons²⁸

In dem Maß, wie einzelne Firmen ihre Produkte unter eine Open-Source-Lizenz stellen (wie etwa Netscape mit Mozilla oder AOL-Server; Zope Corporation mit Zope), wird die Betreuung der Entwickler- und Anwendergemeinde durch eigenes Personal mehr oder weniger notwendig. Dieser Job ist nur in zweiter Linie technisch, es geht mehr um die Moderation der fachlichen Diskussionen, Dokumentation der Produkte und das Management von Versionen und Bug-Fixes. Umfangreiche Produkte (wie Mozilla oder KDE) erfordern sogar mehrere solcher Sprecher, die zwischen der Community und dem zentralen Entwickler des Produkts vermitteln.

5.5.6. Open-Source-Dokumentation

Nicht alle Open-Source-Software ist so gut dokumentiert wie die von PHP oder Apache. Bei manchen Produkten kann es sein, dass die Software nicht oder nur mangelhaft dokumentiert ist. Entscheidet sich ein kommerzieller Anwender oder Berater für dieses Produkt, kann das schnell zu einem Hindernis werden. Statt darauf zu warten, dass Freiwillige einspringen, kann der Interessent jemanden dafür bezahlen, diese Dokumentationsarbeit zu erbringen und Beiträge von anderen in ein "offizielles" Dokument zu integrieren. Die Versuchung ist sicher groß, dieses Dokument /nicht/ zu veröffentlichen, um einen Wettbewerbsvorteil zu haben. In der Praxis ist es aber besser, eben das doch zu tun, denn sobald eine Dokumentation existiert, übernimmt die Community große Teile des Aufwandes der Pflege (ganz ähnlich wie bei Open-Source-Software selbst). Der Umstand, dass es um Open-Source-Software geht, verlangt extra Qualifikationen vom Ausübenden. Zum einen erfordert die Interaktion mit der Community „Soft-Skills“, zum anderen ist viel Information im offenen Quellcode selber versteckt, der interpretiert und enträtselt werden muss. Herkömmliche Dokumentationsarbeit besteht mehr in der rein textlichen Redaktion der Beschreibungen der Entwickler.

²⁸ FdUK: „Der Markt für diese Arbeitnehmer könnte relativ klein bleiben, daher dürfte diese Nische quantitativ wohl eher uninteressant für das AMS sein.“

6.1. Einsatzmöglichkeiten von OS-Software außerhalb des AMS an öffentlichen „Access Points“ oder als spezielle Client Software

Open-Source-Betriebssysteme werden als Grundlage für „Information-“ oder „Internet-Appliances“ immer beliebter. Diese Geräte zeichnen sich dadurch aus, dass sie nur ein einziges Computerprogramm ausführen und als eingekapselte Geräte angeboten werden (etwa als Kassensysteme, vernetzte Waagen, POS-Systeme, etc.) Vom technischen Blickwinkel aus ist dieser Trend zu begrüßen und wird vermutlich proprietäre Lösungen (Windriver, OS/9) nach und nach verdrängen. Der flächendeckende Einsatz von „AMS-Kiosken, d.h. öffentlich zugänglichen Geräten, die nichts als ein Frontend zu den Job-Angeboten des AMS bieten, wäre vermutlich eine sehr durchschlagskräftige Lösung. Der Arbeitssuchende müßte kein spezielles Programm aufrufen, hätte sofort die Suchmaschine zur Verfügung und hätte dank dieser Spezialisierung des Geräts wenig Angriffsfläche für Verwirrung oder Fehlbedienung. Gleichzeitig wäre die psychologische Hemmschwelle vermieden, es mit einem Computer oder einem Betriebssystem zu tun zu haben - der Kiosk würde mehr wie ein Bankomat oder Telefon auftreten.

Diese Vorzüge sind vom Einsatz von Open-Source-Software (Linux als Betriebssystem, eine gesonderte AMS-Kiosk-Applikation als einzige Anwendungssoftware) natürlich unabhängig, legen ihren Einsatz aber nahe. Einerseits wäre die Entwicklung einer solchen Software vergleichsweise gering, und die proprietären Lizenzprodukte erfordern in vielen Fällen hohe Gebühren, so dass sie neben den Entwicklungskosten nicht vernachlässigbar wären. Zum anderen könnte das Offenlegen des Applikations-Codes die Grundlage für andere Arten von Kiosken führen, mit allen üblichen Vorteilen, die daraus erwachsen - höhere Zuverlässigkeit, Gleichstellung von neuen und bisherigen Lieferanten, etc. Da so ein AMS-Kiosk naturgemäß über das Internet vernetzt sein müßte, empfiehlt sich der Einsatz eines frei zirkulierten Betriebssystems (*BSD, Linux, eventuell RT-Linux/Hardhat).

6.2. Klärung der Voraussetzungen zur Optimierung eines offenen Datentransfers (AMS - private Betriebe)

Das bisher über OSS gesagte befasst sich in ganz allgemeinen Worten mit ihren Eigenschaften und ihrem Einfluss auf die Arbeitswelt und die Wirtschaft. Hier nun ein mehr oder weniger willkürlich herausgegriffenes konkretes Beispiel für die Vorteile freier Software und des Open-Source-Prozesses im e-Government: Das Arbeitsmarktservice (AMS).

Es ist wünschenswert und praktisch, die Jobangebote des öffentlichen Arbeitsvermittlers und die der privaten zusammenzuführen und so einen Daten-Pool zu erzeugen, aus dem alle schöpfen können. Dieses Vorhaben erfordert ein gemeinsames, standardisiertes Datenformat zur Beschreibung von Job-Angeboten (evtl. auch Bewerbern) sowie ein Protokoll (Prozess), die zwischen den einzelnen Akteuren ablaufen sollen.

Der mögliche Nutzen hat vier Aspekte.

1.) AMS

Das AMS erhält Besetzungsmöglichkeiten von zusätzlichen offenen Stellen, darunter in Marktsegmenten, in denen das AMS schwach vertreten ist. Die Verweildauer von Arbeitssuchenden wird verkürzt, das statistische Material über den Arbeitsmarkt wird umfangreicher und bildet die Situation besser ab.

2.) Private Arbeitsvermittler

Private Arbeitsvermittler erhalten Zugriff auf das Reservoir an Arbeitssuchenden und Jobs, die beim AMS vorgemerkt sind und erweitern so ihren Aktionsradius.

3.) Potentielle Arbeitgeber

Potentielle Arbeitgeber müssten ihr Inserat nur bei einem einzigen Akteur schalten und hätten die Gewissheit, dass ihr Inserat den Weg zu zahlreichen Job-Börsen und -Vermittlern findet.

4.) Arbeitssuchende

Reicheres Angebot an offenen Stellen, raschere Beendigung der Arbeitslosigkeit, größere Übersicht über den Arbeitsmarkt.

So ein Projekt könnte mit wenig Aufwand begonnen werden und sich zunächst auf die Festlegung des Datenformats und die freiwillige Kooperation einiger privater Job-Börsen beschränken, die dann nach eigenen Vorstellungen die entsprechende Software zur Transaktion von Job-Angeboten und Arbeitssuchenden entwickeln. Es ist auch möglich, diese Technik zunächst auf eine einzige Branche anzuwenden. Bewährt sich dieser Versuch, hätte das AMS eine aussichtsreiches Open Source-Projekt zu vergeben, das eminenten Nutzen stiften könnte und für verhältnismäßig geringe Kosten zu implementieren ist.

7.1. Positionierung des Themas im Rahmen des EU Programms e-Europe

Das Programm e-Europe hat die Entwicklung und Förderung von Informationstechnologien zum Gegenstand und als Schwerpunkt die Förderung von Open-Source-Software in der Verwaltung und der Industrie. Der *Aktionsplan 2005* sieht juristische Maßnahmen, Aufklärung und Subventionen vor, allerdings betreffen nicht alle den Einsatz von OSS, sondern allgemeinere Themen wie elektronische Beglaubigungen, e-Government und Innovationsförderung. Open-Source spielt im Programm e-Europe aber insofern eine Rolle, als dass es klar unterstellt, dass Open-Source-Software alle Belange des Programms fördern und begünstigen kann. Open-Source liefert den idealen Nährboden für Innovation, ist gleichbedeutend mit dem für e-Government geforderten offenen Protokollen und Dateiformaten und macht von einzelnen Herstellern unabhängig.

7.2. Möglichkeiten des Einsatzes von OS-Software im Bereich der öffentlichen Verwaltung

In der öffentlichen Verwaltung wird Software im heutigen Sinne seit den 50er Jahren eingesetzt und ist dort unentbehrlich geworden. Die relativ neue Entwicklung des Open Source-Modells eröffnet drei Möglichkeiten des Einsatzes.

1.) Anwendung von Open Source-Produkten

Die Vorzüge von Open Source Software kommen in der öffentlichen Verwaltung genauso zum Tragen wie in der Wirtschaft. Der Einsatz von horizontalen Produkten wie Linux, MySQL, PostgreSQL, apache, php, etc. bewährt sich bereits in unzähligen Fällen, in denen diese Produkte in öffentlichen Verwaltungen eingesetzt werden (auch in Österreich). Die Vorteile sind geringere Kosten, mehr Flexibilität, höhere Zuverlässigkeit.

2.) Entwicklung von Open Source Software

Die einzelnen Sparten der öffentlichen Verwaltung benötigen für die Computerisierung ihrer Prozesse hochspezialisierte Software, hochspezialisierte Datenformate und hochspezialisierte Software-Komponenten, die zu fertigen Produkten zusammengefügt werden können. Diese Software wird von der öffentlichen Hand bezahlt und könnte (mit Anpassungen) auch in der Wirtschaft wiederverwendet werden, was die

Kosten für Unternehmen reduzieren würde. Gleichzeitig ermöglicht die freie Zirkulation die Weiterentwicklung durch verwandte Institutionen im In- und Ausland, die dem ursprünglichen Investor in diese Software nützt.

3.) Nutzung des Open Source-Prozesses

Bei der Entwicklung internationaler offener Standards für e-Government kann der Open Source-Prozess helfen, den Prozess zur Festlegung der Standards zu beschleunigen. Sobald eine Regierungsstelle ein Datenformat, Protokoll oder Software geschaffen hat, die in der Praxis funktioniert (sie müßte nicht einmal gut funktionieren), wären alle Stellen mit ähnlichen Anliegen eingeladen, dieses Artefakt nach eigenen Vorstellungen zu erweitern, was die Bildung einer Community zur Folge hätte. Diese Community standardisiert die verwendeten Formate oder Protokolle, zirkuliert Software und diskutiert Verbesserungen, bis sich entlang der Praxis, funktionierender Prototypen und dem Konsens der kompetentesten und engagiertesten Mitglieder ein Standard herauskristallisiert. Dieser Prozess ist eine durchschlagskräftige Alternative zu den Normkartellen, Gremien und Ausschüssen, in denen oft politische oder wirtschaftliche Faktoren Einfluss auf die Details eines Standards haben.

7.3. Erfahrungen und Strategien von OS-Software im Öffentlichen Bereich in Deutschland, Frankreich, Spanien, England, Belgien

Open-Source-Software für die öffentliche Verwaltung wird in ganz Europa, sowohl auf EU- als auch auf Mitgliederebene sehr ernst genommen. Das Lobbying dagegen ist sehr engagiert, obwohl es bereits zahlreiche Erfolge und entschlossene Maßnahmen gibt, die verbindliche Auflagen hinsichtlich des Einsatzes von Open-Source-Software einführen sollen. Diese verbindlichen Auflagen gibt es noch nicht. Jedoch setzen einzelne Behörden und staatliche Stellen Open-Source-Software bereits seit mehreren Jahren erfolgreich ein. Deutschland zeichnet sich unter den Mitgliedsländern insofern aus, als dass es Open-Source-Software nicht nur für die Verwaltung fordert, sondern bereits jetzt den breiten Einsatz in der Wirtschaft überhaupt fördert (Informationsbroschüre, Info-Abende für „den Mittelstand“, etc.)

Open-Source-Software in Deutschland:

Deutschland gilt als Europas führendes EU-Mitglied beim Einsatz von Open-Source-Software in der Verwaltung. Es existieren sehr starke Kräfte innerhalb der einzelnen Behörden, ein für die gesamte Verwaltung verbindliches Regelwerk zum Einsatz von Open-Source-Software herzustellen. Unter den Anwendern von Open-Source-Software finden sich praktisch alle Ministerien und staatlichen Stellen, wenn auch oft nur in einzelnen Fällen Open-Source-Software auf Betreiben individueller Fachkräfte Verwendung findet. Eine sehr prominente Nachricht im Mai 2003

war die über die Umstellung von Münchens Stadtverwaltung von Windows NT auf Linux. Obwohl Microsoft zahlreiche Zugeständnisse machte, ließ sich die Stadtverwaltung von diesem Plan nicht abbringen.²⁹

Wenn es um die Koordination der einzelnen Open-Source-Einsätze geht, so gibt es für die deutschen Administrationen keinen ersichtlichen Grund mehr zu warten. Vermutlich wird es sehr bald einen juristischen Rahmen und Richtlinien geben.

Kein Material gibt es über die Vorzüge des Open-Source-Prozesses zur Schaffung von offenen Standards und den Datenaustausch zwischen einzelnen staatlichen Stellen.

Open-Source-Software in England

England liegt beim Einsatz von Open-Source-Software nach eigenen Angaben im Mittelfeld, für den Einsatz von Open-Source-Software in der öffentlichen Verwaltung fehlen Angaben, aber das Thema tritt nicht besonders prominent in Erscheinung; britische Beiträge bei IDA sind eher die Ausnahme [die britische Studie Analysis of the Impact of Open-Source-Software von QinetiQ ist aber sehr treffend und stichhaltig].

Open-Source-Software in Frankreich

Frankreich gehört zu den führenden Open-Source-Nationen der Erde, besonders was den Anteil von Franzosen in Entwickler-Communities betrifft. Beim Einsatz von Open-Source-Software in der Verwaltung ragt Frankreich weit weniger heraus, obwohl sich zahlreiche Stellen darum bemühen und - wie in Deutschland - an einem entsprechenden Regelwerk gearbeitet wird und der breite Einsatz in der Wirtschaft gefördert werden soll.

7.4. Ein kurzer Aufriss der Unterschiede im OS Entwicklungsbereich USA - EU - China/Taiwan

1.) Open-Source-Software in den USA

Die Vereinigten Staaten geben der Open-Source-Welt wichtige Impulse, ihre Dominanz ist aber nicht annähernd so ausgeprägt wie im Bereich proprietärer Software. Es ist richtig, dass die wirtschaftliche Nutzung von Open-Source-Software in den USA besser entwickelt ist (AOL, Red Hat, etc.), und dass auch zahlreiche Open-Source-Produkte von Amerikanern ins Leben gerufen wurden, aber weder bei der Anzahl der Projekte noch beim technischen Niveau der Produkte haben die USA einen ausgeprägten

²⁹ FdUK: „Gerade die Diskussion in Deutschland zeigt, dass die Frage „OSS oder proprietäre Software“ oft nicht ausschließlich auf einer rationalen sondern auch auf einer normativen bzw. emotionalen Ebene geführt wird. Letztlich entscheidend für Beschaffungsentscheidungen sollten in allen Fällen betriebs- und volkswirtschaftliche Kosten-Nutzen-Analysen sein (die allerdings in einigen Fällen auch mit bestimmten Unschärfen behaftet sind). Eine inhaltliche Beurteilung der Münchner Entscheidung kann hier mangels interner Daten allerdings nicht getroffen werden.“

Vorsprung. Auf der anderen, nicht-technischen, Seite wirken die typischen amerikanischen wirtschaftlichen Qualitäten - besser organisierter Kapitalmarkt, engere Verflechtung von Universitäten und Industrie, großzügigere Mittel für die Forschung - auch positiv auf die Entwicklung von Open-Source-Software.

Ob die USA beim Einsatz von Open-Source-Software einen entsprechenden Vorsprung haben, läßt sich ohne aufwändige Untersuchung nicht sagen, aber die Beobachtung der Berichterstattung und der Diskussion läßt folgenden Schluss zu. Die amerikanischen Entscheidungsträger scheinen das größere Vertrauen in proprietäre Hersteller zu haben.

Beim Einsatz von Open-Source-Software im e-Government gibt es zahlreiche Initiativen, die aber auf mehr oder weniger private Initiative von Entscheidungsträgern zurückgehen. Was in den USA ganz fehlt, sind Regierungsbeschlüsse auf Bundesebene, die erwägen, Open-Source-Software selbst einzusetzen oder deren Verbreitung zu fördern (wie es sie in China oder der EU sehr wohl gibt). Die Einführung von OSS in der öffentlichen Verwaltung ist in den USA zur Zeit kein Thema, darüber hinaus existieren rechtliche Rahmenbedingungen - „Intellectual Property - Gesetz“ - die den Einsatz von OS beschränken. Daneben haben zahlreiche Interessensvertretungen bereits sehr restriktive Intellectual Property-Gesetze durchgesetzt, die Open-Source-Software (und auch Open Content, "Open Knowledge") sehr schaden können.

In anderen Worten: Der technische Abstand zwischen Europa und den USA ist auf dem Gebiet der Open-Source-Software wesentlich geringer als auf dem Gebiet der proprietären Software. Open-Source-Initiativen wie die der IDA (EU) fehlen. Zahlreiche Versuche, Open-Source-Software in staatlichen Stellen zu institutionalisieren, sind in den USA gescheitert. Diese Überlegungen sind wichtig, denn sie könnte helfen, sich durch Förderung von Open-Source-Software, bereits jetzt eine Stärke Europas, gegenüber der USA in eine günstigere Wettbewerbsposition zu bringen. Ob die zu Grunde liegende Beobachtung stichhaltig ist, müsste gesondert untersucht werden.

2.) China und Taiwan

Taiwan setzt Open-Source-Software schon seit längerer Zeit ein als China; da es etwas offener und industrialisierter ist, verbreitete sich vor allem Linux früher und rascher. Taiwan hat heute aber im wesentlichen Chinas Initiative zu Open-Source-Software übernommen.

Die Volksrepublik China nimmt Informationstechnologie im Allgemeinen und Open-Source-Software im Besonderen sehr ernst, da sich die chinesische Regierung von Open-Source-Produkten wirtschaftliche Impulse und Verbesserungen der Konkurrenzfähigkeit verspricht. Tatsächlich entwickelten chinesische Universitäten im Auftrag des Staates eine eigene

Linux-Distribution namens Red Flag Linux³⁰, die auch außerhalb Chinas zum Verkauf angeboten wird. Die Regierung bewirbt Open-Source-Software sehr aggressiv und empfiehlt ausdrücklich, Open-Source-Produkten den Vorzug gegenüber "amerikanischer Monopol-Software" zu geben, und diesem Rat wird in immer breiterem Rahmen Folge geleistet. Dieser Umstand ist erstaunlich, denn die meisten chinesischen Anwender erwerben ihre Software nicht, sondern benutzen Raubkopien von proprietären Produkten und chinesische „Raubübersetzungen“ von amerikanischer Fachliteratur. Obwohl Open-Source-Software in China nicht billiger ist als proprietäre Produkte (beide können für den eigenen Gebrauch einfach kopiert werden) erfreut sich die Red Flag-Distribution großer Beliebtheit.

China ist auch sehr unverblümt beim Erklären der Motive dieser Anstrengungen und sämtliche Sprecher vertreten die vorgegebene Parteilinie. So wird ideologisch die Forcierung von Open-Source u.a. damit gerechtfertigt, dass Open-Source als „Vergesellschaftung von Software“ verstanden wird. Die offiziellen Statements zum Open-Source-Kurs der chinesischen Regierung verlaufen entlang der folgenden Argumente.

- o) Open-Source-Software hat hohe Qualität und kostet keine Lizenzgebühren
- o) Open-Source-Software ist extrem flexibel
- o) Open-Source-Software steht für Zukunftssicherheit, Ausfallsicherheit und Einbruchsicherheit
- o) Auf erkannte Mängel in Open-Source-Software wird von der Community sehr schnell reagiert
- o) Open-Source-Software fördert die Kreativität und Innovation in der Industrie
- o) Open-Source-Software ist an die wirklichen Bedürfnisse und Anliegen des Marktes angepasst

Das Anführen dieser Vorzüge ist nicht weiter originell, denn sie gehören zu den klassischen Argumenten für Open-Source-Software. Interessanter sind da einige Zeilen von Liu Bo, Gründer und Präsident von Red Flag Software. Er ist ein typischer Sprecher für die chinesische Open-Source-Parteilinie.

Laut Liu Bo fällt Linux in China auf sehr fruchtbaren Boden, denn:

- o) die chinesische Regierung fördert den Einsatz
- o) Chinesen können sich die hohen Lizenzgebühren proprietärer Software nicht leisten
- o) Unix war immer sehr beliebt in China
- o) China wird bald der größte Telekom-Markt der Erde sein
- o) China ist der größte Markt für elektronische Geräte und Information Appliances [die dann Linux als Software enthalten sollen]

³⁰ FdUK: „Möglicherweise würden hier in vielen westlichen Unternehmen (aus vielleicht auch nur subjektiven Sicherheitserwägungen) Bedenken gegen den Einsatz von „Red Flag Linux“ bestehen (Angst vor Backdoors, Industriespionage,...).“

Als "politische" Motive führt Liu Bo unter anderen an:

- o) Informationssicherheit bedeutet nationale Sicherheit
- o) Das Risiko wird vermieden, dass Information durch fremde Instanzen kontrolliert wird
- o) "Don't [sic] like Microsoft monopoly"

Als wirtschaftliche Gründe führt Liu Bo an:

- o) "XOO Millionen Dollar" Lizenzgebühren entfallen
- o) Die Kompetenz der lokalen Software-Industrie steigt

Weder die Motive der chinesischen Regierung noch die Schlussfolgerungen sind originell oder überraschend. Besonders viel verspricht sich die chinesische Regierung vom Aufbau von Software-Kompetenz und -Know-how durch Open-Source-Software. Erklärter Konkurrent im Standortwettbewerb ist Indien, dessen Regierung die Förderung von Open-Source-Software ebenfalls forciert.

7.5. Lokale Wertschöpfung und Auswirkungen auf die Produktivität³¹

1.) Mehr Produktivität durch „Monopolisierung“

Obwohl einzelne Open-Source-Produkte bzw. Projekte miteinander in Konkurrenz treten, kristallisiert sich in der Regel sehr schnell ein Gewinner heraus, der dann die Ressourcen und oft Teile des Codes der Mitbewerber assimiliert. Das findet überall dort statt, wo zwei Projekte etwa das selbe Terrain abdecken und mehr oder weniger austauschbar sind. Oft kommt es in so einem Wettstreit auch zur Spezialisierung, d.h. die Konkurrenten erkennen ihre jeweils eigenen Vorzüge und konzentrieren sich auf die Weiterentwicklung dieser Vorzüge (wie etwa der superkompakte und minimalistische Editor vi und der barocke, enorm flexible und programmierbare emacs). Für die Gesamtproduktivität bedeutet diese „Monopolisierung“, dass Produkte nur einmal entwickelt werden. Sämtliche Leistungsmerkmale oder Verbesserung fließen danach in dieses eine Produkt ein. Eine Innovation erfordert nicht, mit dem Marktführer in Wettstreit zu treten, indem um die Innovation ein komplettes Produkt entwickelt wird. Stattdessen fließt die Innovation in die bestehende Software ein. Obwohl Open-Source-Software enorm competitive ist, wird der Konkurrenzkampf zwischen Ideen ausgetragen, nicht zwischen Produkten, die kollaborativ entwickelt werden. Da sämtliche Entwickler an nur einem Produkt arbeiten und nicht an mehreren, für die Arbeit dupliziert werden muss, ist die Entwicklung von Open-Source-Produkten im Gesamten billiger.

2.) Mehr Produktivität durch Forschung und Entwicklung

³¹ FdUK: „Zu diesem Thema ist gerade eine ergänzende umfangreiche und detaillierte wissenschaftliche Studie in Arbeit. Daher werden die Ausführungen dieses Kapitels an dieser Stelle nicht näher kommentiert.“

Vor der weiten Verbreitung von Open-Source-Software standen experimentierfreudige Grundlagenforscher vor einigen gewaltigen Problemen, ihren Projekten vielfach die Freude und die Glaubwürdigkeit nahmen. Beispielsweise erforderten neue Ansätze bei Dateisystemen, Datenbank-Algorithmen, Scheduling, etc. oder neue Hardware-Plattformen die Schaffung von Prototypen, deren Praxistauglichkeit oft nur in Prototypen demonstriert werden konnte, weil die Software fehlte, auf die portiert werden konnte, oder in die Komponenten integriert werden konnten. Heute ist die Portierung von Linux, zusammen mit der gesamten Tool-Chain, auf neuartige Hardware-Plattformen eine typische Übung in Universitäten und bei Chip-Herstellern. Neuartige Applikationen, Betriebssystembestandteile oder Software-Metriken werden zuerst auf, an, mit, um, oder in Linux getestet und demonstriert und stehen ab da als Ergänzung zur Verfügung. In anderen Worten: Statt Prototypen zur Verwendung in Produkten zu verbreiten, wird der Prototyp sofort in das Produkt eingestrichelt und zum Produktbestandteil. Neue Prozessortypen oder andere Hardware erscheinen heutzutage mit praktisch dem gesamten Reservoir an Open-Source-Software, die dank Emulatoren schon lauffähig ist, noch bevor die erste Hardware die Fabrik verlässt. Sehr viel Aufwand ist in die Untersuchung des Linux-Codes nach formalen Kriterien geflossen, die von Wissenschaftlern entwickelt und an Linux erprobt wurden. Dieses Anzapfen von Grundlagenforschung und dem Reservoir an Diplomanden hat Open-Source-Software in der Vergangenheit viel genutzt und wird in Zukunft für die rasche Verbreitung von wichtigen Innovationen sorgen. Besonders wenn diese Vorgehensweise institutionalisiert ist.

3.) Mehr Produktivität durch Open-Source-Middleware

Open-Source-Software wird im Bereich der sogenannten „Middleware“ proprietäre Produkte vermutlich nach und nach verdrängen. „Middleware“ sind Halbfertig-Produkte, die sich durch Konfiguration, Anpassungen und Programmierung in spezialisierte Produkte verarbeiten lassen. Im Augenblick sind besonders im kaufmännischen Bereich viele proprietäre Produkte keine Middleware, sondern auf bestimmte Sparten und für die Verwendung in bestimmten Ländern spezialisiert. Solche Software könnte durch Open-Source-Middleware insgesamt wirtschaftlicher hergestellt werden, weil sie einander stark ähneln, aber die gemeinsamen Teile nicht gemeinsam genutzt werden können, weil der Quelltext nicht offengelegt ist. Die Hersteller dieser Software stiften den Nutzen durch Kombination von selbst entwickelten und mehr oder weniger austauschbaren Komponenten mit spezialisiertem Branchen- und kaufmännischem Know-how. Die Folge sind hohe Kosten.

Ein Beispiel: Software zur Verwaltung eines Videoverleihs in Österreich unterscheidet sich der Natur nach nur wenig von der zur Verwaltung eines spanischen Videoverleihs. Die Steuersätze, staatlichen Auflagen und lokalen Gegebenheiten mögen sich geringfügig unterscheiden, aber diese Unterschiede rechtfertigen im Grunde genommen nicht zwei lokale Produkte von zwei verschiedenen Herstellern. Die freie Zirkulation des Quelltextes

derartiger Produkte würde es einer breiten Palette von lokalen Anbietern ermöglichen, eine große Zahl von einander sehr ähnlichen Produkten zu geringen Kosten zu schaffen. Ähnliche Überlegungen gelten für jede Form von spezialisierten Produkten für industrielle, medizinische oder administrative Anwendungen. Die erhöhte Produktivität ergibt sich als Folge der Offenlegung des Quelltextes - nur jene Teile, die eine bestimmte Applikation gegenüber einer anderen auszeichnen, müssen neu entwickelt werden. Die gleichen Teile werden einfach aus bestehenden Bibliotheken übernommen.

Die Maßnahmenvorschläge gliedern sich in drei Bereiche: in Arbeitsmarkt und Bildung, in Wirtschaft sowie die öffentliche Hand. Im Vordergrund steht dabei die Umsetzung; in welchem Bereich, muss sicher noch diskutiert werden.

8.1. Arbeitsmarkt und Bildung

8.1.1) Schule

8.1.1.1.) Forcierter ergänzender Einsatz von Open-Source-Produkten in der Erstausbildung (Beispiele: Linux und KDE/Gnome statt Microsoft Windows, Mozilla statt Internet Explorer, StarOffice statt Microsoft Office).

8.1.1.2.) Bildungsziel Technologie-Bewertung
Ein attraktives Bildungsziel ist differenzierte Sicht auf Technologie im Allgemeinen und ihre Auswirkungen auf die Gesellschaft. (Beispiele für konkrete Themen: Digital Divide, Copyright, Patent-Gesetzgebung.)

8.1.1.3.) OSS-Wettbewerb zwischen Schulen.
Gute Resultate und Anregungen können auch für "richtige" Projekte genutzt werden (Forschung, Initiativen).

8.1.2.) Arbeitsmarkt - Arbeitsmarktausbildung

8.1.2.1.) Kursangebote nach Bedarf vermehrt auch an Open-Source-Software, aber auch Verwendung von spartenspezifischen Produkten (z.B. Gimp für Graphiker, Photographen und Designer). Verwendung und Analyse von Linux-Software, um prinzipielle Belange der Informatik zu illustrieren und zu vermitteln.

8.1.2.2.) Vermittlung von projekt- und prozessorientierten Lehrinhalten
Produktion von fachspezifischem Content wie Benutzerhandbücher für die verwendeten Geräte, Zusammenstellung von "Best Practices"-Sammlungen, etc. Daneben Peer Review, kollaboratives Arbeiten, Verwertung der Erkenntnisse und Arbeiten aus vorangegangenen Kursen.

8.1.2.3.) Einrichtung von "virtuellen Lernzentren"
Mit Open Content. Ein virtuelles Lernzentrum bietet Internet-Zugang und Zugriff auf Texte, Animationen und anderes Material zur Vermittlung dieser Inhalte. Didaktisch ausgebildete Fachleute sollten in die Gestaltung der Vermittlungsprozesse mit einbezogen werden (Planungsboards,

Qualitätssicherung). Daneben sollten auch Trainer und Coaches zur Verfügung stehen.

8. 1. 2.4.) Förderung von Tele-Working³²

Nach dem Modell der Lehrlingsförderung könnten Unternehmer Förderungen für Tele-Arbeitsplätze bekommen.

8.1.2.5.) Förderung von Soft-Skills

wie Teamarbeit, Selbstvermarktung, Kommunikation, Krisenmanagement

8.1.2.6.) Lehrlingsoffensive in Richtung „Open Source Worker“. Praxisnahe Berufsausbildung für Lehrlinge beispielsweise als sog. LAMP Entwickler (Linux/Apache/MySQL/Php), eine der meist verwendeten Produktionsumgebungen für KMU Web- und e-Business Sites. Der Umfang einer solchen Ausbildung würde in den Zeitrahmen einer Lehrlingsausbildung passen und es gäbe sicher viele Betriebe die solche Lehrlinge ausbilden/Einstellen würden.

8.2. Die Wirtschaft

8.2.1.) KMU-Informationskampagne

durch Veranstaltungen, Broschüren, Inserate, Website (z.B.: <http://www.open-source.at>), etc..

8.2.2.) Einrichtung von Open-Source Consulting³³

Nach dem Vorbild der Jungunternehmerberatung könnte man spezielle Anlaufstellen für KMU einrichten, bei denen sich Unternehmer für den Einsatz von Open-Source-Produkten beraten lassen können.

8.2.3.) Schaffung einer KMU-Distribution

Viele österreichische Unternehmer haben sehr ähnliche Anforderungen (FiBu, Steuer, etc.) und könnten eine Linux-Distribution gebrauchen, die speziell für ihre Bedürfnisse zusammengestellt wurde und sich "Out-of-the-Box" installieren lässt.

8.2.4.) Staatliche Zertifizierung von Open-Source-Software³⁴

Als vertrauensbildende Maßnahme sollte es ein staatlich beglaubigtes Zertifikat für Open-Source-Software geben, um bestimmte Produkte für bestimmte Zwecke als "sehr gut geeignet" zu etikettieren.

³² FdUK: „Bei den prinzipiell begrüßenswerten neuen Formen der Arbeit müsste in jedem Fall sichergestellt sein, dass es zu keiner sozialen Schlechterstellung der Arbeitnehmer kommt.“

³³ FdUK: „Es gibt ein Programm des WIFI mit neutraler geförderter Beratung von Unternehmen durch gewerbeberechtigte Unternehmensberater („UNS“). Dieses Programm könnte z.B. auch für IT-Beratung übernommen werden und müsste ebenfalls neutral sein.“

³⁴ FdUK: „Diese Zertifizierung müsste dann im Sinne der Förderung der Markttransparenz für alle Arten von Software zur Verfügung stehen und nicht nur auf OSS beschränkt sein. Das Zusammenspiel mit internationalen Zertifikaten (z.B. Sicherheit) müsste geklärt werden, um eine Verwirrung der Benutzer zu vermeiden.“

8.2.5) Einrichtung einer Open-Source Businessplattform

Unternehmer und öffentliche Stellen, die Projekte auf Basis von Open-Source-Software realisieren möchten, sollen hier eine Anlaufstelle haben bei der sie das Projekt vorstellen können.

Es sollte Foren zu verschiedenen Themen des OSS Einsatzes in Wirtschaft und eGovernment geben. Mittels einer „Experten-Datenbank“ und einer Anbindung an das AMS sollten auch geeignete Projektteams zusammengestellt werden können.

8.3. Die Öffentliche Hand

8.3.1.) Unterstützung und Forcierung von OS-Think Tanks

Förderung von Innovation bedeutet, auch radikal neue Ideen zu fördern, deren unmittelbarer Nutzen nicht offensichtlich bzw. absehbar ist, und die ein großes Risiko bergen, niemals oder erst nach langer Zeit profitabel zu werden.

8.3.2.) Erweiterung des "Kommunikationsplans IKT"

Es wäre wertvoll, eine möglichst breite Öffentlichkeit um Maßnahmen wie die hier aufgelisteten wissen zu lassen, zusammen mit den Motiven, Zielen und Erfolgsmeldungen.

8.3.3.) Definition der Standards für Software im öffentlichen Bereich

E-Government wird eine große Zahl von Standards erfordern, um die Integration von Daten zwischen einzelnen Stellen zu ermöglichen.

8.3.4.) Nationales und europäisches Steuerungsgremium für regionale, nationale und³⁵ internationale Koordination von großen Software-Projekten. Zur Optimierung der Nutzung der Synergien.

8.3.5) Vergabe von Projekten durch die öffentliche Hand

Zum Aus- und Aufbau einer IT - Infrastruktur könnte die Öffentliche Hand entsprechende Infrastruktur-Projekte - antizyklisch - an die IT- Wirtschaft vergeben. Die Konjunktur belebende Maßnahmen sollten vor allem auf die Bedürfnisse und Möglichkeiten von KMU-IT Unternehmen Rücksicht nehmen.

³⁵ FdUK: „Da Softwareentscheidungen auf den Einzelfall zugeschnitten sein sollten und betriebswirtschaftlichen Rentabilitätsüberlegungen folgen sollten, müsste die Einhaltung des Subsidiaritätsprinzips gewährleistet sein.“

Anhang 1: Referenzen

Allgemein

Softwarestudie 2000

Eine empirische Untersuchung der österreichischen Softwarebranche

<http://www.wi.wu-wien.ac.at/~bernroid/swstudie2000/>

Univ. Assistent DI Edward W.N. Bernroider und MMag. Walter Ebner

Ergebnis einer neu konzipierten und umfangreicheren Folgeuntersuchung einer bereits im Jahre 1992 veröffentlichten Marktstudie.

.....

Free/Libre and Open-Source-Software: Survey and Study

<http://www.infonomics.nl/FLOSS/>

Erhebung über die sozio-ökonomischen Auswirkungen von Open-Source-Software.

.....

http://www.govtalk.gov.uk/interoperability/egif_document.asp?docnum=430

Eine Untersuchung der britischen Regierung, wie Open-Source-Software den Standort UK beeinflussen könnte.

.....

Open Collaboration and Standards in e-Government

Open-Source: A Case for E-Government

<http://www.infodev.org/presentations/OpenSource02/niemann.pdf>

Eine Untersuchung der Weltbank über die möglichen Vorzüge von Open-Source-Software in der öffentlichen Verwaltung (der USA).

.....

Umfrage der EU: Open-Source in der Verwaltung

<http://news.berlios.de/article.php?sid=351>

bzw.: <http://www.cio.gv.at/alternativen/ida-oss/>

Die IDA Initiative (Interchange of Data between Administrations) der EU-Kommission hat vor kurzem eine Studie über die Verwendung von Open-Source-Software in der öffentlichen Verwaltung veröffentlicht (Study into the use of Open-Source-Software in the public sector).

.....

Open-Source-Software in Deutschland (Überblick, Einsatz)

<http://europa.eu.int/ISPO/ida/export/files/en/605.pdf>

Egon Troles

Koordinierungs- und Beratungsstelle (KBSt)

Kompetenzzentrum für Informationstechnik und Kommunikation der Bundesregierung für Informationstechnik in der Bundesverwaltung im Bundesministerium des Innern, Bonn

.....

Eine Empfehlung der IDA (Interchange of Data between Administrations)

<http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&parent=news&documentID=550>

Open-Source-Software in der öffentlichen Verwaltung.

.....

POOLING VON OPEN-SOURCESOFTWARE

<http://europa.eu.int/ISPO/ida/export/files/de/1320.pdf>

IDA-Machbarkeitsstudie

Datenaustausch zwischen Verwaltungen

Europäische Kommission, GD Unternehmen

.....

Warum "Open-Source"?

Deutschland auf dem Weg in die Informationsgesellschaft

<http://ig.cs.tu-berlin.de/w2002/osp/001/Uberblick1-endg.pdf>

Dr. Ulrich Sandl, Bundesministerium für Wirtschaft und Technologie,

sandl@bmwi.bund.de

.....

http://www.firstmonday.org/issues/issue8_1/schweik/index.html

Ein eher theoretischer Essay über Open-Source-Software, und wie sie Standardisierungsprozesse im allgemeinen und auch im öffentlichen Sektor fördern kann.

http://e-economy.berkeley.edu/conferences/9-2000/EC-conference2000_papers/salmelin.pdf

Eine Untersuchung der Universität Berkeley zum Thema e-Commerce, gute Daten, eine quantitative und qualitative des Standorts Europa auf dem Feld des e-Commerce.

.....
<http://www.linux-magazin.de/Artikel/ausgabe/2000/08/TagungTutzing/TagungTutzing.html>

Linux-Magazin: Freie Software - Ein Modell für die Bürgergesellschaft?

.....
Freie Software und Bildung e.V
<http://fsub.schule.de/indexf.htm>

Verein zur Förderung von allem, was dem Verständnis und dem selbst bestimmten Umgang mit den neuen Informations- und Kommunikationstechnologien dient. (Auf Basis von OSS)

.....
Stiftung digitale Chancen
<http://www.digitale-chancen.de/content/sections/>

Im Angebot des Netzwerks Digitale Chancen verbindet sich die Kompetenz einer Vielzahl von Experten unterschiedlicher Fachrichtungen mit der von Vertreterinnen und Vertretern aller gesellschaftlichen Gruppen.

.....
Politik Digital
<http://www.politik-digital.de/>

Politik Digital ist ein Verein 'für eine demokratische und digitale Entwicklung der europäischen Informationsgesellschaft' mit Hauptsitz in Berlin.

.....
SPLITTER 1/2001
<http://www.lit.berlin.de/BVC/splitter/sp1-2001/spinh1-2001.htm>

Der SPLITTER informiert (nicht nur) die Berliner Verwaltung über alle Themen der Datenverarbeitung.

.....
Computer Professionals for Social Responsibility
<http://www.cpsr.org/>

Providing the public and policymakers with realistic assessments of the power, promise, and problems of information technology. CPSR members work to direct public attention to critical choices concerning the applications of information technology and how those choices affect society.

.....

FITUG e.V. Förderverein Informationstechnik und Gesellschaft
<http://www.fitug.de/>

Förderung der Integration der neuen Medien in die Gesellschaft, die Aufklärung über Techniken, Risiken und Gefahren dieser Medien, sowie die Wahrung der Menschenrechte und der Verbraucherschutz in Computernetzen. Durch die genannten Zwecke sollen Kultur, Bildung und Wissenschaft gefördert werden.

.....

Akademiker gesucht
<http://www.heise.de/tp/deutsch/html/result.xhtml?url=/tp/deutsch/inhalt/co/13829/1.html&words=Arbeitsmarkt>

Die Zahl der Studenten erreicht einen neuen Höchststand - trotzdem gehen die deutschen Hochschulen schweren Zeiten entgegen...

.....

Nichts mehr mit Funny Money (Ulrich Hottel, 05.04.2001)
<http://www.heise.de/tp/deutsch/html/result.xhtml?url=/tp/deutsch/inhalt/te/7293/1.html&words=Arbeitsmarkt>

New Economy: Genug von rein virtuellen Arbeitnehmerrechten

OSS in China/Taiwan

Dieses Dokument hat auf der Seite 10 das im Text erwähnte offizielle Statement von Yun Peng Wang, Präsident des Management Institutes im Ministry of the Industry Information:
<http://www.bull.com/techtrends/opensource.pdf>

Hier die im Text erwähnten Folien zum Vortrag von Liu Bo, Gründer von Red Flag Linux: http://www.iandiasia.com/presentation/Liu_Bo.ppt

Ein interessanter Artikel, der belegt, dass die chinesische Regierung extrem kompetent bei der Auswahl und Reihenfolge der Maßnahmen zur Förderung der eigenen IT-Industrie durch Open-Source-Software ist:
<http://www.open-mag.com/1743583279.htm>

Ein für den Gebrauch am Desktop optimiertes Linux namens Yangfan-Linux vom „Beijing Software Industry Productivity Center“, einer Regierungsstelle

zur Koordination der Weiterentwicklung von Linux:

<http://www.theregister.co.uk/content/4/26737.html>

Ein Artikel über Microsofts Lobbying in Shanghai, China - Geldspenden, Joint Ventures: <http://news.com.com/2100-1001-932927.html>

Dieser Artikel erzählt, dass Microsoft in China wegen Open-Source-Software in der Defensive ist:

<http://news.bbc.co.uk/2/hi/science/nature/1749441.stm>

"China has launched an ambitious plan in software industry recently trying to catch up with India by year 2006."

http://galletta.business.pitt.edu/amcis2003/05INTERNATIONAL3_China.htm

Hier eine Lagebeurteilung der Open-Source-Situation in China durch einen deutschen Sinologen und Linux-Kenner:

<https://www.linux->

[magazin.de/Artikel/ausgabe/2000/04/ChinaLinux/chinalinux.html](https://www.linux-magazin.de/Artikel/ausgabe/2000/04/ChinaLinux/chinalinux.html)

Anhang 2: Glossar

Administrator

Ein Administrator im hier verstandenen Sinne managt einen oder mehrere Maschinen (Server) in einem Computer-Netzwerk. Die genaue Tätigkeit hängt von der Software ab, die auf diesen Maschinen läuft (Mail-Server, Web-Server, Datenbank-Server, etc). Von einem Administrator großer Netzwerke oder komplexer Anwendungen wird hohe Qualifikation und Motivation gefordert.

AMS

Arbeitsmarktservice

AMS-Kiosk

Das AMS erwägt, öffentlich zugängliche Computer in "Kiosken" einzurichten, die als Terminals für den Zugriff auf die Jobangebote des AMS auftreten.

Ausfallsicherheit (Stability)

Computer und Computerprogramme sind berüchtigt für "Abstürze", die durch defekte Hardware oder Programmierfehler in laufender Software verursacht werden. Diese Ausfälle zu vermeiden, erfordert viel Aufwand und durchdachte Software ohne Programmierfehler.

Benutzerergonomie

Computerprogramme werden immer umfangreicher und sollten so bequem wie möglich zu bedienen sein. Als "Benutzerergonomie" bezeichnet man heute das Ausmaß, in dem eine bestimmte Software eine durchdachte und den Aufgaben angemessene Benutzer-Schnittstelle hat.

Betriebssystem

Die elementare Software, die die rohe Hardware überhaupt zum Leben erweckt. Ein Betriebssystem vermittelt in genau spezifizierter Weise zwischen der Elektronik der Maschine und den Computerprogrammen, die darauf laufen.

Bibliothek

Einer der Schwerpunkte beim gekonnten Programmieren ist das Wiederverwenden von Code. Bestimmte Unterprogramme können in vielen verschiedenen (aber mehr oder weniger spezialisierten Zusammenhängen) genutzt werden, wenn sie sinnreich und flexibel genug erstellt sind. Eine Sammlung von solchen Unterprogrammen heißt Bibliothek („Library“); viele solche Sammlungen gibt es am Internet (in der Regel Open Source). Daneben gibt es auch Hersteller von proprietären Bibliotheken, die die Nutzung je nach Exklusivität und Qualität ihrer Produkte vergewähren. Beispiele für Bibliotheken: Die „clib“ (Teil der Standard-Programmiersprache C) für elementare Ein/Ausgabe-Aufgaben; die objektorientierte MFC (gehört zum

Lieferumfang von Microsofts Entwicklungsumgebung „Visual Studio“) für die Programmierung von Windows-Applikationen; OpenGL von CGI zum dreidimensionalen Zeichnen von Objekten, ihren Oberflächen und ihren Schatten. Bei vielen Open Source-Projekten fallen Bibliotheken als Nebenprodukt zur Verwendung für andere Projekte an (asyncore/Medusa), die dann bei überlappendem Funktionsumfang miteinander in Wettstreit stehen. Gelegentlich treten die konkurrierenden Communities miteinander in Verbindung, um ihren jeweiligen Code durch Entwicklung einer gemeinsamen Bibliothek zu konsolidieren (wie bei „Parrot“, der gemeinsamen „virtuellen Maschine“ für python und perl).

Boom-Jahre des Internet

Nach den ersten Erfolgen des World Wide Web (ca. 1993) wurde eine breite Öffentlichkeit auf die Möglichkeiten des Internet aufmerksam und investierte im Überschwang mehrere hundert Milliarden Dollar in völlig neuartige Internet-Firmen. Es gab sehr viele derartige Firmen und sehr viele Investoren, was andere Märkte (Medien, Hardware, Marketing, Betriebsberatung, Telekommunikation, etc.) mitriss. Dieser Boom dauerte von ca. 1998 bis 2000. Ob der wirtschaftliche Aufschwung und besonders die Hausse überhaupt ökonomische Grundlagen hatten, wird heute lebhaft diskutiert.

Brain-Drain

Das Abwandern von qualifizierten Fachkräften in eine andere Region, in der sie bessere Bedingungen vorfinden.

Business Process Software

Alle Software, die im Unternehmen die Abläufe implementiert (Logistik, Lager, Buchhaltung, Korrespondenz, etc.)

Business Productivity Software

Ein allgemeiner Sammelbegriff für typische Büro-Anwendungen wie Textverarbeitung, Tabellenkalkulation, e-Mail, Kalender, Graphik-Editor. Diese Software zeichnet sich dadurch aus, dass ein Individuum an seinem Arbeitsplatz damit interagiert -- im Gegensatz zu Software, die "unsichtbar" bestimmte Services über das Netzwerk anbietet.

Code

Code im hier verstandenen Sinne ist ein etwas vager Begriff und bezeichnet Teile eines Computerprogramms in - für Programmierer - lesbarer Form. Siehe auch Quelltext oder Source Code.

Commodification

Die Vermarktung und Vergebung einer allgemein verfügbaren Ressource (wie in krassen Fällen Trinkwasser, Müllabfuhr). Commodification wird oft in Zusammenhang mit Privatisierung genannt, ist aber nicht immer dasselbe.

Community

Gemeinschaft, die um ein bestimmtes Anliegen oder eine Zielsetzung herum

aufgebaut ist. Hier wird Community im Sinne der Gemeinschaft verstanden, die um die Verbreitung des Open-Source-Gedankens oder eines bestimmten Open-Source-Produkts aufgebaut ist. Eine Projekt-Community ist in der Regel um die Software selbst aufgebaut, wird aber von einem Projektleiter (siehe dort) moderiert.

Competence Network

Ein soziales, lockeres Netzwerk von Individuen die miteinander in Kontakt stehen und einander bei der Arbeit unterstützen. Diese Unterstützung kann vielfältige Formen annehmen -- vom einfachen, telephonisch erteilten, Ratschlag über Vermittlung potentieller Geschäftspartner bis zur ergänzenden Kooperation in einem Projekt. Wichtig ist hier, dass ein Competence Network unabhängig von Firmen oder Unternehmensstrukturen ist. Teilnehmer unterstützen einander auch oft, wenn sie in miteinander im Wettbewerb stehenden Unternehmen arbeiten.

Content Management System

Spezielle Software zur Erzeugung und Verwaltung von Content, d.h. Magazinartikeln, Fachaufsätzen, Bildern, Leserbriefen, etc. -- "Inhalten", die dann veröffentlicht werden. Ein Content Management unterstützt in der Regel auch "Privilegien", d.h. je nach Rang dürfen nicht notwendigerweise alle Benutzer Artikel erzeugen, verändern oder lesen. Content Management Systeme die über das World Wide Web bedienbar sind, werden zunehmend komplexer, vielseitiger und mächtiger.

Distribution

Im hier verstandenen Sinne bedeutet Distribution ein Paket von Open-Source-Software (typischerweise eine Vielzahl von Linux-Software), die dann praktisch abgepackt und installierbar verkauft oder verschenkt wird.

Dot.Com-Bubble

"Dot.Com" wurde zum Synonym für "Web-Firma". Investoren versprachen sich von solchen Firmen hohe Erträge oder wenigstens hohe Gewinne bei der Spekulation mit Anteilen an solchen Firmen. Das "Bubble" in "Dot.Com-Bubble" ist ein geringschätziger Term; nach der Euphorie des Internet-Booms (dem Platzen der "Seifenblase") zeigte sich, dass die Dot.Coms nicht viel Substanz hatten und das Geld der Investoren verloren war.

Dummie ("Dammi")

Eine durch den IDG-Verlag populär gewordene Bezeichnung für einen IT-technischen Laien, in dem mild-ironische Geringschätzung mitschwingt.

E-Commerce

Vermarktung und Vertrieb von Waren oder Dienstleistungen über das World Wide Web.

E-Government

Computerisierung der öffentlichen Verwaltung; Bereitstellung von Dienstleistungen der öffentlichen Verwaltung über das World Wide Web;

Beglaubigung, Schutz von vertraulichen Daten durch elektronische Schlüssel und Kryptographie.

Einbruchsicherheit (Security)

Die inzwischen zum Folklore gehörende Figur des "Hackers", (eigentlich "Cracker") der sich einer fremden Maschine als Unbefugter bemächtigt, Daten ausspäht, verändert oder zerstört, ist heute so weit verbreitet, dass sie wohl keiner Erklärung mehr bedarf. Solche Einbrüche werden entweder durch sicherheitsrelevante Programmierfehler möglich oder durch inkompetente Wartung des kompromittierten Programms oder Maschine. Einbruchsicherheit ist daher ein wichtiger Anspruch an die verwendete Software, besonders, wenn sie über das Internet öffentlich zugänglich ist.

Extranet

Neben dem Begriff Internet haben sich heute "Intranet" und "Extranet" etabliert. Ein Intranet ist ein firmen- oder organisationsinternes Netzwerk, das exakt so wie das Internet funktioniert, aber nur von internen Netzwerk-Teilnehmern verwendet werden kann. Ein "Extranet" ist ein Zusammenschluss solcher Netzwerke, die zwar räumlich getrennt, aber über das Internet verbunden sind. In anderen Worten: Die Teilnehmer der einzelnen Intranet-Inseln sind über das Internet verbunden, können miteinander kommunizieren und erscheinen einem einzigen Netzwerk zugehörig; Außenstehende haben keinen Zugriff auf die einzelnen Inseln. Ein Synonym für "Extranet" ist "Virtual Private Network".

Horizontal

Hier wird "horizontal" im Sinne der Betriebswirte verstanden: Keiner bestimmten Sparte zugehörig, nicht spezialisiert, generisch.

Hosting

Eine Dienstleistung, die die Bereitstellung von Ressourcen für die Speicherung von Daten oder für den Betrieb von Computerprogrammen beinhaltet. Hosting kann die bloße Vermietung von Plattenspeicher, kompletten Maschinen, Benutzerkonten oder Rechenzeit für bestimmte Programme umfassen.

Hypertext Transfer Protocol

Das genormte "Protokoll" über das einzelne Programme (oder Maschinen) im World Wide Web kommunizieren. Dieses Protokoll ist ein Standard für das Format, in dem Daten abgepackt sein müssen, um als Web-Daten entziffert werden zu können.

Information-Appliance

Ein elektronisches Gerät, das nur einen einzigen oder nur wenige Zwecke erfüllt (z.B. Abspielen von MP3s, Mobiltelefonie, Web-Surfing, Text-Verarbeitung, etc.) -- im Gegensatz zu einem PC, der jedes Computerprogramm laufen lassen kann.

Internet-Appliance

Eine elektronisches Gerät, das nur einen einzigen oder wenige Zwecke erfüllt,

die das Internet erfordern (Internet-Telephonie, Internet-Radio, Web-Surfing, etc.)

Internet Engineering Task Force

Eine gemeinnützige Organisation, die technische Standards für das Internet durch groben Konsens ("rough consent") festlegt. Die IETF ist eine NGO und basiert auf freiwilliger Teilnahme. Es steht jedem offen, Vorschläge zu machen oder zu diskutieren. Die IETF veröffentlicht die Spezifikationen für Standards in sogenannten "Request For Comments" (RFCs), die nummeriert sind und dann bestimmte Datenformate, Protokolle oder Prozeduren normen.

Intranet

Siehe Extranet.

IT-Infrastruktur

Eine bestimmte flächendeckende elementare Dienstleistung (im weitesten Sinne), auf der andere (Infrastruktur)leistungen aufgebaut werden können. Das Straßennetz und seine Pflege ist eine wichtige Infrastrukturleistung, auf der dann die Müllabfuhr, ebenfalls eine wichtige Dienstleistung aufgebaut ist. Das Telephonnetz ermöglicht unter anderem Notrufe, Telefonseelsorge, Pizza-Lieferungen. Bei der IT-Infrastruktur ist das ähnlich - ein weitverbreitetes Betriebssystem, die Vernetzung der einzelnen Maschinen, grundlegende Anwendungen, etc. ermöglichen es einem Individuum oder Unternehmen, mit IT zu arbeiten und Mehrwert zu schaffen. In diesem Text wird IT-Infrastruktur nicht nur im Sinne öffentlicher Dienstleistungen verstanden (Internet, Linux, Windows, digitale Signatur, etc.), sondern auch in dem Sinne, dass jedes Unternehmen über seine eigene IT-Infrastruktur verfügen muß; (Intranet, spezialisierte Anwendungen, Server für bestimmte Aufgaben, e-Mail, Datenbanken, etc.)

Lebenszyklus (einer Technologie)

Eine neue Technologie beginnt ihr Leben in der Regel als Idee oder als Prototyp, der einer bestehenden Technologie Konkurrenz macht (wie der Kraftwagen Pferd und Kutsche). Diese Prototypen sind meistens mitleiderregend und werden zunächst nicht als Nachfolger der bestehenden Technologie gesehen - oft ist zunächst auch nicht absehbar, dass der Prototyp überhaupt irgendeine praktische Anwendung hat. In diesem Stadium wendet sich eine eng begrenzte Avantgarde dem neuen „Spielzeug“ zu. Diese Avantgarde konstituiert sich aus Visionären, die das Potential der neuen Technologie erkennen; Unternehmern, die durch Zufall in ihrer Nische sofort unmittelbaren Nutzen daraus ziehen können (Jahrmarkt-Kuriosität); oder (in der Regel vermögenden) Liebhabern der neuen Technologie. Mit der Weiterentwicklung der Technologie dringt sie zunächst in einige Nischen vor und/oder löst dort bestehende Technologie ab (Militär, Luxusgut, Sportwagen), bis sie reif für den Massenmarkt (Henry Ford, Modell T) ist und ihr Potential für alle erkennbar ist. Gleichzeitig kommt eine Spirale in Gang, bei der sich externe Faktoren an die neue Technologie angepaßt werden (Ausbau des Straßennetzes, Tankstellen, Werkstätten, etc.), was sich mit der

Verbreitung aufschaukelt. Gleichzeitig tritt Spezialisierung ein (Lastwägen, Kombis, Kühlwägen, Motorräder, Sportwagen, Pritschenwagen, etc.) und die Technologie-Fauna und -Flora wird immer facettenreicher. An diesem Punkt ist die abgelöste, einst etablierte Technologie ihrerseits in Nischen zurückgedrängt (Fiaker, Sportereignisse). Während wir bei Kraftfahrzeugen noch lange warten müssen, bis sie das Schicksal ihrer Vorgänger-Technologie ereilt, blicken wir bei Midrange-Computern („Mini-Computern“) auf diesen kompletten Lebenszyklus zurück. Mini-Computer begannen als kostspielige, aber erschwingliche Alternative zu Großrechnern und wurden zunächst von Universitäten zur Forschung und Lehre herangezogen (das Modell T der Mini-Computer war die Digital Equipments PDP-1). Diese Maschinen waren primitiv, reichten aber, um Prinzipien zu verdeutlichen und erste Erfahrungen mit Computern zu machen. Nach und nach wurden die Maschinen besser und mächtiger, andere Hersteller stiegen in das Geschäft ein, und die Absolventen begannen, die vertrauten Systeme in der Industrie und im Gewerbe einzuführen. Digital Equipment spezialisierte sich auf hochwertige, extrem durchdachte und zuverlässige Maschinen und wurde Marktführer in diesem Segment. Der Markt für Software für mittlere Unternehmen wurde lebendig und evolvierte sehr rasch in Richtung netzwerkfähiger Software und Dienstleistungen. Der Arbeitsmarkt war ebenfalls hochspezialisiert, die Ausübenden waren gesuchte Spezialisten (VAX VMS, PDP-11 Assembler, Fortran-77, BLISS). Ab 1980 wurde für Visionären absehbar, dass Microcomputer eines Tages den Platz der Mini-Computer einnehmen würden, was zu diesem Zeitpunkt weniger weitsichtigen Zeitgenossen nicht recht plausibel vorkommen wollte. Seit 2000 sind Mini-Computer verschwunden, es werden keine neuen Systeme mehr entwickelt (die meisten werden nicht einmal mehr gewartet); nur in Bereichen, in denen man nicht die bewährte Infrastruktur durch Einführung neuer Systeme aufs Spiel setzen will. Der Arbeitsmarkt für die damals verwendeten Programmiersprachen und Betriebssysteme ist praktisch verschwunden.

Interessant in diesem Zusammenhang sind aber zwei Dinge: Erstens, dass Unix, ein klassisches Betriebssystem für Mini-Computer, erst nach dem Untergang dieser Hardware-Kategorie seinen Aufschwung erlebte. Unix-Experten, die ihre ersten Erfahrungen bereits mit Mini-Computern machten, genießen trotz oder wegen ihres Alters hohes Prestige und sind eine begehrte Rarität. Zweitens: Die Mini-Computer-Landschaft war gewissermaßen die Blütezeit der proprietären Standards und proprietären Systeme, die untereinander völlig inkompatibel waren. Jeder Hersteller, versuchte, seinen Claim durch Geschlossenheit abzustecken. Unix und das Internet-Protokoll TCP/IP sind hier die Ausnahmen und die wenigen Überlebenden aus der Midrange-Ära - noch dazu sehr erfolgreiche. Es ist gewiß kein Zufall, dass gerade Unix und TCP/IP offene Standards sind und, wenn man den Begriff sehr weit faßt, immer Open Source Software waren.

Linux

Das wohl prominenteste Open-Source-Produkt, das dem Gedanken und der Bewegung seinen Durchbruch verschafft hat. 1991 entwickelte Linus Torvalds -

ein finnischer Informatik-Student - einen Betriebssystemkern namens Linux. Da der Open-Source-Bewegung gerade so ein Kern fehlte, um ihr gigantisches Reservoir an freier Software zu einem eigenständigen, kompletten Betriebssystem vereinigen zu können, entstanden nach kurzer Zeit Sammlungen von diversen Open-Source-Programmen plus Betriebssystem (sog. Distributionen, siehe dort).

Lizenz

Im hier verstandenen Sinne: Software-Lizenz. Lizenzen für proprietäre Software verbriefen das Recht, für den Preis der Software die Software nutzen zu können. Lizenzen für Open-Source-Software verbrieft in der Regel das Recht, die Software uneingeschränkt nutzen, verändern und weitergeben zu können. Einige Lizenzen (wie Linux' General Public License) auferlegen dem Benutzer aber, 1.) den Source Code vollständig weiterzugeben, nicht nur das lauffähige Programm, und 2.) sämtliche Verbesserungen öffentlich zu zirkulieren, die man vornimmt.

Micro-Enterprise

Kleine Firmen, die Ihre Verbindungen, Ressourcen und Know-how so geschickt einsetzen können, dass sie gewaltige Umsätze praktisch ohne fixes Personal, ausgedehnten Firmensitz oder Kapital erzielen. Micro-Enterprises konzentrieren sich auf die Organisation und die Vermarktung von Projekten.

Migration

Die Umstellung von Software oder IT-gestützten Prozessen auf ein anderes Betriebssystem bzw. andere Software.

Offener Standard

Ein Standard, dessen Details genau bekannt sind, weil er in öffentlich zugänglichen Dokumenten genau spezifiziert ist. Proprietäre Standards sind oft nicht dokumentiert, das heißt, die genaue Codierung von Daten ist nur dem Hersteller bekannt. Beispiele für offene Standards: Das Web-Protokoll HTTP, das Rich Text Format RTF.

Offenlegung des Quelltexts

Der Quelltext, der "Code" (siehe dort), ist die Blaupause eines Computerprogramms, und im Gegensatz zum lauffähigen Programm, von Fachleuten lesbar und verstehbar. Da Quelltext gewissermaßen das Betriebsgeheimnis des Herstellers enthält, wird er bei proprietärer Software in der regel nicht frei zirkuliert. Der zentrale Gedanke hinter Open-Source-Software ist, eben das zu tun, um jedem die Möglichkeit zu geben, ein bestimmtes Produkt zu verstehen und zu verändern.

Open-Source-Produkt

Ein Software-Produkt, das jeder zusammen mit dem Code (siehe dort) weitergeben und verändern darf. Siehe Offenlegung des Quelltexts.

Open-Source-Prozess

(auch Open-Source-Modell.) Open-Source-Software wird von Freiwilligen geschrieben, die ihre Produkte nach eigenen Vorstellungen und Anliegen schaffen oder bestehende Produkte ergänzen. Der Open-Source-Prozess ist nicht hierarchisch oder zentralistisch organisiert, sondern "organisch" -- inkrementelle Verbesserungen werden von jenen gemacht, die sie benötigen; die eigene Arbeit baut auf der anderer auf; Probleme und Differenzen werden durch Diskurs und Fachautorität gelöst; funktionierende Prototypen, lauffähige Programme und Common Sense entscheiden diese Diskussionen.

Open-Source-Software

Software, die jeder zusammen mit dem Code (siehe dort) weitergeben und verändern darf. Siehe Offenlegung des Quelltexts.

Outsourcing

Die Überantwortung eines Geschäftsbereiches (wie Fertigung, Kunden-Support, etc.) an einen externen Dienstleister. Hier wird Outsourcing als Beauftragung eines externen Dienstleisters mit IT-Aufgaben verstanden (Website, Wartung des Netzwerks, Entwicklung von spezieller Software).

Point-And-Click

Die Einführung der Maus und graphischer Benutzeroberflächen ist einer der bahnbrechenden Leistungen der IT, die die Bedienung von Computern dramatisch erleichtert. "Point-And-Click" wurde zur Bezeichnung für das Arbeiten mit Maus und einer graphischen Benutzeroberfläche.

Programmiersprache

Computerprogramme enthalten exakte Angaben an die Maschine, was in welcher Reihenfolge zu tun ist (etwa wie ein Kochrezept). Diese Anweisungen sind in einer Programmiersprache formuliert, die für Fachleute verstehbar ist, aber zu komplex und zu fremdartig, um unmittelbar von der Hardware ausgeführt zu werden. Spezielle Computerprogramme sind notwendig, um diese für Menschen bequemen Formulierungen zu interpretieren und in eine ausführbare Form zu bringen.

Proprietärer Standard

Im Gegensatz zu offenen Standards (siehe dort).

Proprietäre Software

Proprietäre Software ist Eigentum des Herstellers, der Kunde erwirbt mit der Lizenz nur das Recht zur Nutzung. Der Käufer darf die Software in der Regel nicht kopieren, darf keine Änderungen vornehmen, und jeder Versuch, herauszufinden, wie die Software funktioniert ist strafbar. Das alles dient dazu, das geistige Eigentum des Herstellers zu schützen. Im Gegensatz dazu steht Open-Source-Software (siehe dort), die all diese Rechte ausdrücklich verbietet und sogar fördert.

Protokoll

Im hier verstandenen Sinne bedeutet Protokoll einen festgelegten Ablauf, nach

dem zwei Maschinen über ein Netzwerk kommunizieren. Das Protokoll legt beispielsweise fest, unter welchen Bedingungen die Kommunikation abgebrochen, in welcher Weise sie begonnen oder beendet wird. Protokolle sind in der Regel standardisiert (eine Norm) oder selbst ein Standard.

Quellcode

Synonym von Code (siehe dort), Quelltext, Source, Source Code, etc.

Red Flag Linux

Eine Zusammenstellung von Linux-Software von der chinesischen Regierung, die in China zum offiziellen Betriebssystem und die Alternative zu Microsoft-Produkten werden soll.

Renaissance der Mainframes (Großrechner)

In den 80ern, unter dem Eindruck der raschen Zunahme der Mächtigkeit von PCs, wurden die unverhältnismäßig kostspieligen Großrechner ("Mainframes") totgesagt.

Nach einem dramatischen Einbruch in den Verkäufen und dem Untergang zahlreicher Mainframe-Hersteller konsolidierte sich aber der Markt, und die verbliebenen Hersteller konzentrieren sich seither auf die Stärken ihrer Großrechnerprodukte: hohe Ausfall- und Einbruchsicherheit, Speicherung gewaltiger Mengen von Daten, hohe Leistung im Datendurchsatz, große Anzahl von gleichzeitig ablaufenden Prozessen. Besonders die Betreiber großer Websites, so stellte sich heraus, sind mit Großrechnern besser bedient als mit verteilten (PC)-Lösungen. Diese "Renaissance der Mainframes" wird in absehbarer Zeit zu einem Mangel an entsprechend ausgebildeten Fachleuten führen und wurde zu einem Beispiel für Probleme, die durch das voreilige Abschreiben einer Technologie entstehen können.

Sabbatical

Aus dem amerikanischen Universitätswesen stammende Praxis für Mitarbeiter in Forschung und Lehre, die dann auch von amerikanischen High Tech-Firmen übernommen wurde: nach einigen Jahren intensiver Tätigkeit nimmt der Ausübende ein Jahr bezahlten (unbezahlten oder teilweise bezahlten) Urlaub, um sich weiterzubilden oder an eigenen Projekten zu arbeiten, die mit seiner bezahlten Tätigkeit nicht unmittelbar zu tun haben. Die Abmachung lautet in der Regel, dass die Erkenntnisse dieser Tätigkeit vom Arbeitgeber verwertet werden dürfen.

SAP

Der prominenteste Hersteller von Business Process Software (siehe dort). SAP sieht sich aber nicht als Lieferant von Software-Paketen, sondern als Dienstleister, der Betriebsberatung macht und Abläufe im Unternehmen des Klienten auf die SAP-Software umstellt.

Software-Fehler

Programmierfehler. Software-Fehler sind praktisch unvermeidbar und ihre

Vermeidung und Behebung macht einen bedeutenden Teil der Kosten der Software-Entwicklung aus.

Software-Metrik

Ein Verfahren zur Beurteilung der Arbeit von Programmierern nach formalen Kriterien (Fehlerrate, Anzahl der Zeilen pro Zeiteinheit, Verwendung bestimmter idiomatischer Wendungen, etc.) Software-Werkzeuge Computerprogramme zur Erstellung oder Bearbeitung neuer Computerprogramme.

Source (Code)

Siehe Code.

Tele-Working

Heimarbeit, bei der der Ausübende über ein Netzwerk mit Daten, Ressourcen und Mitarbeitern seines Arbeitgebers verbunden ist.

Tool-Chain

Software wird in der Regel einmal vom Programmierer geschrieben und danach durch automatische Prozesse in Instruktionen umgewandelt, die die Hardware ausführen kann. Diese Prozesse involvieren mehrere spezialisierte Computerprogramme zur Erzeugung dieser Instruktionen, die nacheinander ausgeführt werden. Da der Ablauf solchermaßen "verkettet" ist, heißt die Gesamtheit dieser Programme auch oft "Tool-Chain".

Total-Cost-of-Ownership

Die Anschaffungskosten für Hardware und Software sind nicht die einzigen, die während des Einsatzes anfallen. Die Wartungskosten, Schulungskosten und versteckten Kosten durch Ausfälle oder Sicherheitsprobleme übersteigen jene für die Anschaffung in der Praxis bei weitem. Die Gesamtheit all dieser Kosten wird als "Total-Cost-of-Ownership" bezeichnet.

Total-Cost-of-Ownership-Diskussion

Seit proprietäre Software-Hersteller in Open-Source-Produkten ernstzunehmende Mitbewerber sehen, weisen sie oft darauf hin, dass Open-Source-Produkte zwar Anschaffungskosten von Null haben, aber im Betrieb eine höhere Total Cost of Ownership (siehe dort) haben als proprietäre Software. Es gibt zwar Studien, die diese These belegen, in jüngster Zeit überwiegen anderslautende Studien.

Trustworthy Computing

Eine Initiative von Microsoft, die PCs sicherer machen sollen. Die Methoden zum Erreichen dieses ehrgeizigen Ziels sind noch unklar, werden aber spezielle Hardware und Software (Palladium, Passport, etc.) erfordern. Kritiker befürchten, dass die Technologie dazu missbraucht werden wird, um Mitbewerber aus dem Wettbewerb auszuschließen.

Unix

Unix ist ein 1969 von Brian Kernighan und Ken Thompson entwickeltes Betriebssystem, das viele Innovationen beinhaltet, die heute noch ihren Wert haben. Die gesamte Unix Systemphilosophie umfasst zahlreiche Konzepte und Komponenten, die dann im Laufe der Jahrzehnte bis heute verfeinert wurden und gewissermaßen das heute bekannte Optimum eines Betriebssystems seiner Klasse konstituieren. Seit seiner Entwicklung wurden zahlreiche Unix-Clones geschaffen, die das ursprüngliche Unix mehr oder weniger exakt nachbilden, und die ihre jeweiligen Verbesserungen sehr rasch voneinander abgeschaut haben. Auch Linux ist eine solcher Unix-Clone.

Virenepidemie

Die enge Integration einzelner Applikationen wie e-Mail, Textverarbeitung und Programmierung dieser Applikationen unter Microsoft Windows ermöglicht Computerprogramme, die mit der e-Mail verschickt oder in Text-Dokumente eingebaut sind und ohne die Erlaubnis des Benutzers gestartet werden. In Kombination mit der sehr laxen Security unter Microsoft Windows ist es möglich, Programme zu schreiben, die sich selbst über das Netzwerk verschicken und sich so sehr schnell verbreiten. Diese "Virenepidemien" umfassen meistens den gesamten Erdball und können beträchtlichen Schaden anrichten. Sie sind aber auf Windows-kompatible Maschinen beschränkt.

Web-Service

Ein Service, der über das World Wide Web bereitgestellt wird. Wichtig ist hier, dass der Begriff nicht nur solche Services bezeichnet, die von Menschen in Anspruch genommen werden, sondern auch solche, die zur Weiterverarbeitung anderen Maschinen weitergereicht werden (über HTTP, SOAP, etc.)

Zertifizierungsindustrie

Zahlreiche Hersteller proprietärer Software bieten Kurse und Zertifikate für die Kenntnis ihrer Produkte an, die in der Wirtschaft hohes Ansehen genießen (Novell Certified Network Engineer, Cisco Certified Network Engineer, Microsoft Certified Network Engineer, etc.) Um diese Zertifikate herum hat sich eine spezialisierte und weitverzweigte Industrie gebildet, die Schulungen, Fachliteratur, Testsimulationen und Zeitschriften anbietet, um auf die Prüfungen vorzubereiten.

Teil A 3

Index

- Abläufe, nicht-formalisierte 15
- Access Points 42
- Administrator 60
- Akquisition von Wissen 13
- Aktionsplan 2005 44
- alternative Technologie 21
- AMS 42, 43, 54, 60
- AMS-Kiosk 42, 60
- Anforderungen der Wirtschaft 38
- Angebot und Nachfrage von IT-Fachkräften 8
- Anreiz für Konzentration 12
- Antizyklisch 54
- Anwendung von Open Source-Produkten 25
- AOL 18
- AOL Server 18
- Apache 15, 17, 20, 25, 28, 29, 33, 44
- Apple OS X 32
- Arbeitsvermittler, private 43
- Arbeitgeber 25
- Arbeitgeber, potentielle 43
- Arbeitgeber, potentielle 43
- Arbeitsmarkservice 28
- Arbeitsmarkt 52
- Arbeitsmarktausbildung 52
- Arbeitsorganisation 19
- Arbeitsstätte 30
- Arbeitssuchende 43
- Arbeitsvermittler, private 43
- Arbeitswelt 25, 37
- Arbeitszeit 30
- Argumente für den Einsatz von Open Source Software 10
- Artefakte, technische 20
- Aus- und Weiterbildung 37
- Ausbildung von Informatikern 38
- Ausbildung, praxisorientierte 38
- Ausfallsicherheit 18, 48, 60
- Auswirkungen der Konjunktorentwicklung 9
- Automatisierung 11, 21
- Automobilindustrie 11
- Beharrungsvermögen der IT-Nutzung 9
- Belgien 45
- Bell Labs 15
- Benutzerdokumentation 19
- Benutzerergonomie 11, 60
- Benutzerfreundlichkeit 18
- Berkley Unix 25, 32
- Berufsbezeichnungen, neue 39
- Beschäftigungsmodell 30
- Best Practices 27, 52
- Betriebssystem 60
- Bibliothek, objektorientierte 26
- Bildungsinstitute 28
- Bildungskarenz 30
- Bildungsträger, etablierte 38
- BIND 17, 35
- Boom-Jahre des Internet 7, 61
- Brain-Drain 61
- BSD (Berkley Unix) 17
- Bürgernähe 21
- Business Process Software 11, 61
- Business Productivity Software 61
- Business-to-Business 11
- C++ 26
- CCITT 23
- China 48
- Cisco Certified Professional 27
- Cisco-Router 32
- Clisp 17
- CMS-Entwickler 40
- Coaches 53
- Code 19, 20, 30, 61
- Commodification 61
- Community 13, 17, 18, 20, 27, 29, 45, 62
- Community-Liasons 40
- Compentence Networks 31
- Competence Network 62
- Competence-Network 30
- Computer-Integrated Manufacturing 11
- Computerprogramm 42
- Computerprogramme 20
- Content 21, 52
- Content Management System 62
- Copyright 52
- Datenaustausch 46
- Datentransfer, offener 42
- de facto-Standard 26
- Delphi 9
- Denken, komplexes 38
- Deutschland 45
- Die 11
- Die Kathedrale und der Basar 30
- Dienstleister 12
- Dienstleistungsgewerbe 11
- Differenzierung des IT-Marktes 12
- Digital Divide 52
- Diplomanden 50
- Distribution 62
- Diversifizierung 27
- Diversifizierung der Ausbildungsmöglichkeiten 37
- dot.com-Bubble 7, 25

Dot.Com-Bubble 62
 Dummy 62
 e-Commerce 63
 e-Commerce-Techniker 38
 e-Europe 44
 e-Government 22, 23, 24, 43, 44, 45, 63
 Einbruchssicherheit 18, 48, 63
 Einsatz von Open Source-Software in der Arbeitsvermittlung 42
 elektronische Beglaubigungen 44
 Emacs 17
 emacs Editor 50
 Emacs Editor 33
 England 45
 Enthusiast 20
 Entwickler 30
 Entwicklung von Software 25
 Entwicklungsmodell, hierarchisch 19
 Erfahrungen, einschlägige, der Länder der Europäischen Union 44
 Erfinder 22
 Erfolgsmeldungen 54
 Eric S Raymond 11
 Erwachsenenbildung 28, 38
 EU-Ebene 21
 Europa 45
 Europas Stärke 47
 Experten-Datenbank 54
 Extranet 63
 Fachhochschule 28
 Fachkräfte 13
 Fachkräfte, Mangel 29
 Fachkräftemangel in der IT-Branche 7
 Fachliteratur 29
 Fachmann 29
 Fachverlag 19
 Festlegung des Datenformats 43
 Filmbranche 31
 Filmwirtschaft 31
 Finanzamt 21
 Fixpersonal 31
 Flexibilität 29
 FLOSS 10
 Forschung und Entwicklung 50
 Frankreich 45, 46
 FreeBSD 32
 freier künstlerischer Beruf 30
 Frontal-Modell 14
 Gedankenexperiment 24
 Geschäftsprozesse 14, 15
 Gesellschaft 20
 gesellschaftliche Zusammenhänge 20
 Gewerbeansiedlung 12
 GIMP 17
 Glaubwürdigkeit 21, 50
 Gleichgesinnte 29
 GNOME 35
 Gnu Software 32
 Groß-EDV 11, 15
 Großrechner 9
 Großunternehmen 11
 Güterindustrie 11
 Hardhat 42
 Hardware 14, 15
 Hardware-Hersteller 12
 HDTV 23
 Hersteller 13
 Hochschulabschluß in Informatik 13
 Höhere Ausfall- und Einbruchssicherheit 10
 Hollywood 30
 Horizontal 63
 Horizontal (Software) 17
 Hosting 14, 63
 HTL 28
 Hypertext Transfer Protocol 13, 23, 24, 63
 IBM 12, 14, 15
 Industrieansiedlung 12
 Info-Abende 45
 Information Appliances 42, 49
 Information-Appliance 64
 Informationsbroschüre 45
 Informationskampagne (KMU) 53
 Informationstechnologie 11, 12
 Informationstechnologie, China 48
 Informationstechnologien 44
 infrastruktureller Einsatz 10
 Infrastruktur-Projekte 54
 Inhomogenität, regionale 12
 inkrementelle Fortschritte 22
 INN 35
 Innovation 22, 54
 innovative Produkte 22
 Integration von Ideen 22
 Intellectual Property-Gesetze 47
 Internet Appliances 42
 Internet Engineering Task Force 23, 24, 64
 Internet-Administrator 38
 Internet-Appliance 64
 Internet-Foren 20
 Intranet 64
 Ist-Zustand des Arbeitsmarktes 7
 Ist-Zustand des Marktes 7
 IT, industrielle Anwender 15
 IT-Aufgaben, Psychologie und Natur 13
 IT-Ausgaben 9
 IT-Berufe 27, 28, 37
 IT-Betriebsberater 15
 IT-Branche 26, 30, 31, 38
 IT-Branche, 12
 IT-Branche, einheitlicher Arbeitsmarkt 12
 IT-Branche, Repräsentation der Sektoren 12
 IT-Dienstleistungen 13
 IT-Fachkräfte 27, 31, 37
 IT-Fachkräfte, ältere 13
 IT-Fachkräfte, Altersverteilung 13
 IT-Fachkräfte, Angebot und Nachfrage 7

- IT-Fachkräfte, Lebensarbeitszeit 9
- IT-Fachmann 25
- IT-Infrastruktur 10, 12, 14, 15, 64
- IT-Kenntnisse 38
- IT-Kostenfaktoren bei KMU 14
- IT-Nutzung 10
- IT-Professionisten 25
- IT-Services 11
- IT-Spezialisten 25
- IT-Zentren 12
- Japan 23
- Java 26
- Job-Angebote 43
- Job-Börsen 43
- Jungunternehmerberatung 53
- Karriere 28
- KDE 34
- klassische Unternehmen 10
- KMU 11, 16
- KMU-Distribution 53
- Know-How 15, 31
- Knowledge-Worker 30
- KOffice 36
- Kommunikationsfreudigkeit 14
- Kommunikationsplan IKT 54
- Konjunkturschwäche 7
- Konsulent 13
- Kontakte, informelle 31
- Konzentration (von Know-How, Infrastruktur, etc) 12
- Kooperation zwischen Universitäten und Industriezentren 12
- Koordination einzelner Open Source-Einsätze 46
- Kostensparnis 31
- Kostenreduktion durch Laien 29
- Kraftwerksbau 18
- Kreative 21
- Kreativität 28
- Krisenmanagement 53
- Kultur 39
- Kundenwünsche 19
- Labortest 24
- LAMP-Experte 39
- Lebenslanges Lernen 37
- lebenslanges und berufsbegleitendes Lernen 28
- Lebensqualität 30
- Lebenszyklus (einer Technologie) 64
- Lehrbücher 20
- Lehrlingsausbildung 53
- Lehrlingsförderung 53
- Leistungsmerkmal 19
- Lernen, kommunikatives 28
- Liebhabelei 25
- Linux 15, 17, 20, 25, 26, 29, 32, 44, 48, 66
- Linux bin-utils 17
- Linux-Code 50
- Liu Bo 49
- Lizenz 66
- Lizenzgebühren 10, 48, 49
- Lobbying 45
- logistischer Einsatz 10
- Mail-Administrator 38
- Mainstream-Spezialisten 26
- Management, hierarchisch organisiertes 20
- Management-Methoden 31
- Management-Techniken 19
- Marketing-Abteilung 19
- Markt, horizontaler 26
- Marktführer 50
- Marktwert einer Qualifizierung 8
- Markup-Specialist 39
- Mediator 28
- Metriken für Wirtschaftlichkeit und Qualitätssicherung von Open Source S...
- Micro-Enterprise 30, 31, 66
- Microsoft 15, 26
- Microsoft Foundation Classes 26
- Microsoft Windows 26
- Middleware 50
- Migration 25, 66
- Ministerien 46
- Mittelstand 45
- Modell Hollywood 31
- Monopolisierung 49
- Monopol-Software, amerikanische 48
- Motivation 19, 28
- Motive, politische 49
- Mozilla 18, 36
- Münchner Stadtverwaltung 46
- mySQL 33
- MySQL 28, 29
- NetBSD 32
- Netscape 18
- Networking, soziales 31
- neue Selbständige 30
- Nischenprodukt 20
- Nischenprodukte 20
- Novell 9
- Novell Certified Professional 27
- offene Standards 23
- offener Standard 66
- Offenlegung des Quelltexts 51, 66
- öffentliche Hand 14, 54
- öffentliche Verwaltung 44, 45
- öffentliche Verwaltung und Open Source 44
- Office Productivity Software 11
- Open 20
- Open Cores 21
- Open Knowledge 47
- Open Office 11, 35
- Open Source Consulting 53
- Open Source Produkt 67
- Open Source Software 9, 19, 67
- Open Source Software als Alternative zu gängigen proprietären Systemen
- Open Source Software in den USA 47
- Open Source Software in der Verwaltung 4
- Open Source Software in der Wirtschaft 25

- Open Source Software und KMU 11, 15
- Open Source Software, Anwendungsgebiete 17
- Open Source Software, Auswirkungen auf den Arbeitsmarkt 4
- Open Source Software, Bedeutung der Vorzüge 10
- Open Source Software, Einsatzgebiete 17
- Open Source Software, Förderung von 13
- Open Source-Ansätze 28
- Open Source-Äquivalente 25
- Open Source-Bereich 31
- Open Source-Bewegung 32
- Open Source-Businessplattform 54
- Open Source-Dokumentation 40
- Open Source-Experte 37
- Open Source-Lizenz 18, 40
- Open Source-Modell 44
- Open Source-Produkt 13
- Open Source-Produkte 15, 18, 26
- Open Source-Projekte 19
- Open Source-Prozess 20, 22, 31, 45, 67
- Open Source-Servermarkt 25
- Open Source-Technologie 13
- Open Source-Verfahren 13
- Open Source-Welt 13, 26
- OpenBSD 32
- Opportunitätskosten, Fehleinschätzung 9
- OS/9 42
- OS-Fachkräfte 25, 28
- OSS-Technologien 11
- OSS-Wettbewerb 52
- OS-Think-Tank 54
- Outsourcing 14, 67
- Partizipation 22
- Patch-Work-Karrieren 28
- Patentgesetzgebung 52
- PC-kompatible Systeme 11
- Peer Review 52
- perl 17
- Perl 25, 26, 33
- Persistence of Vision 35
- Personal Computer 11, 15
- Personal, billigeres 29
- PHP 25, 34
- physische Ressourcen 12
- Point-And-Click 29, 67
- POS-Systeme 42
- PostgreSQL 33
- Präsenz, räumliche 12
- Praxistauglichkeit 50
- Prestige 20
- Produkt, horizontales 24
- Produktivität 19, 49, 50
- Programmierer, freiwillige 19
- Programmierfehler, Minimierung von 18
- Programmiersprache 19, 26, 67
- Programmiersprache, Kenntnis einer bestimmten 8
- Projektleiter 13
- Projektleiter, Qualitäten eines 13
- proprietär Software 67
- proprietäre Produkte 29
- proprietäre Projekte 19
- proprietäre Software 15, 19
- proprietäre Software-Welt 25
- Proprietärer Standard 67
- Protokoll 44, 68
- Prototyp 24
- Prototypen 50
- Prozesse, 21
- Prozesse, IT-gestützte 14
- Prozessleittechnik 18
- python 17
- Python 26, 34
- QinetiQ 46
- Qualifikation 27, 38
- Qualität 18, 20
- Quellcode 68
- Rationalisierung der Prozesse 10
- Raubkopien 48
- Raum, ländlicher 13
- Raymond, 11
- Raymond, Eric S 30
- Red Flag Linux 48, 68
- Red Flag-Distribution 48
- RedHat 47
- Regelwerk zum Einsatz von Open Source Software 46
- Regierung, chinesische 48
- Renaissance der Mainframes 68
- ruby 17
- Sabbatical 8, 68
- Samba 34
- SAP 12, 15, 69
- Schaltkreise 21
- Schlüsselfigur (eines Projekts) 19
- Security 63
- Selbststudium 13
- Selbstvermarktung 53
- sendmail 15
- Sendmail 25, 28, 33
- Server-Anwendungsbereich 25
- Server-Software 25
- Soft-Skills 53
- Software 14
- Software als Ware 11
- Software, Modularität 17
- Software, programmierbare 17
- Software, Wirtschaftlichkeit 18
- Software-Fehler 69
- Software-Hersteller 12
- Software-Metrik 69
- Softwarepaket, als Ware 11
- Software-Studie 15
- Source (Code) 69
- Spanien 45
- Spezialisierung 27, 37
- Spezialkenntnisse 18, 26

staatliche 21
Stability 60
Stability and Security 10
Stallman, Richard 32, 33
Stand der Technik 22
Standard 13
Standardsicherheit 11
Star Office 11
StarOffice 18, 35
Stellen, staatliche 46
Steuergruppe 5
Steuerungsgremium 54
Sun 14
Taiwan 48
Teamarbeit 53
Teamwork 28
Technik 20
technische 20
technische Zusammenhänge 20
Technologie-Bewertung 52
Telearbeit 30, 31
Tele-Working 53, 69
TeTex 35
Tool-Chain 50, 69
Total-Cost-of-Ownership 69
Total-Cost-of-Ownership-Diskussion 69
Transparenz 21, 22
Transparenz von Open Source Software 21
Tretmühle 37
Trustworthy Computing 70
Tutorials 20, 29
Umstieg 8
Umstieg und Alter 8
Umstieg, Bereitschaft 13
Unisys 14
Universität 12, 28, 37
Universitätsabschluss 38
Unix 11, 17, 70
Unix-Administrator 28
Unix-Implementation 32
Unix-Systemphilosophie 15
Unix-Welt 13
urbane Zentren 12
Urlaubsanspruch 30
USA 23
Vergleich 20
Verschnittoptimierung 18
Verwaltung 21, 22
Verwaltung, öffentliche 22
Verwertbarkeit, unmittelbare 38
vi Editor 33, 50
Vim Editor 33
Virenepidemie 18, 70
virtual prototyping 11
virtuelle Lernzentren 52
Vorreiterrolle des IT-Bereichs 30
Wachstumsrate von Open Source Software 25
Web-Administrator 38
Web-Programmierer 37
Web-Service 70
Web-Site, Gestaltung 14
Werkzeuge, proprietäre 26
Wertschöpfung, lokale 49
Wettbewerbsposition, günstigere 48
Windriver 42
Workshop 6
Xfree86 35
XML-Designer 39
X-Window 35
Zertifizierung 27, 53
Zertifizierungsindustrie 27, 70
Zertifizierungssystem 27, 28
Zertifizierungssysteme für OSS 39
Zope 17, 34
Zopist 39
Zukünftige Entwicklung des IT-Arbeitsmarktes 8
Zukunftssicherheit 15, 48
Zuverlässigkeit 44