

Die offene Wissensgesellschaft und ihre Feinde

Alexander Schatten
<http://www.schatten.info>
email: alexander@schatten.info
27. Februar 2003

Inhaltsverzeichnis

1	Einleitung	2
2	Definition der Begriffe „Open Source“ und „Open Protocol“	2
2.1	Open Source	2
2.2	Open Protocol	5
3	Unterschiede zwischen „offenen“ und konventionellen Entwicklungsprozessen	6
4	Alternativen und Risiken	8
5	Wirtschaftliche Konsequenzen	10
6	Bezüge zu Politik und Gesellschaft	11
7	Dystopische Entwicklungen?	14
7.1	Dystopien	14
7.2	„Erfolgsfaktor“ Wissen als Ressource?	15
7.3	Bedingungen für ökonomischen Erfolg in der Wissensgesellschaft	17
7.4	Konsequenzen für Open Source/Protocol Projekte	19
8	Fazit	20
9	Anmerkungen zu Autor und Dokument	21

1 Einleitung

Die Idee zu diesem Text entstand im Kontext eines wirtschafts-philosophischen Seminars („Globalisierung und Wissen als Bedingung und Möglichkeit einer zeitgenössischen Wirtschaftsphilosophie“) und ist eine Auseinandersetzung mit dem Thema „Open Source (Software) Entwicklung und deren Einflüsse auf Wirtschaft und Politik“. Dieser Artikel ist *work in progress* und wird kontinuierlich erweitert. **Viele der Punkte, die in der aktuellen Version angesprochen werden, sind nur angedacht und werden in weiterer Folge auch mit der Hilfe anderer Autoren, die in Spezialgebieten über besseres Wissen verfügen, in mehr Details analysiert werden.** Im besonderen gilt dies auch für die aktuellen Entwicklungen um TCPA und Palladium.

Zunächst werden die recht technischen Begriffe *Open Source* und *Open Protocol* erklärt, danach folgt eine Auseinandersetzung mit den Einflüssen der Open Source/Protocol Bewegung auf Wirtschaft und Politik, sowie die möglichen Gefahren die von Monopolen bzw. Gesetzgebung drohen. Anschließend folgt ein kurzer Querverweis auf andere im Seminar angesprochene Themen, v.a. auf den Text „Dystopia“ von Helmut Willke [20].

Besonderer Dank gilt Hannes Kirschbaum, der diesen Text aus einer nicht-technischen Perspektive gelesen hat, und viele wesentliche Korrekturen sowie Verbesserungsvorschläge angebracht hat.

2 Definition der Begriffe „Open Source“ und „Open Protocol“

2.1 Open Source

„Die Krisis des Wissens wird, wie gesagt, kognitiv getrieben von der neuen Relevanz des Nichtwissens, und sie wird operativ davon getrieben, dass es nun darum geht, die richtigen Fehler schneller zu machen als die Wettbewerber, um Lernprozesse zu intensivieren, die im Kern darin bestehen, Expertise im Umgang mit Nichtwissen zu entwickeln.“ *Helmut Willke [20]*

Open Source ist der in der allgemeinen Rezeption verglichen mit dem Begriff *Open Protocol* der wohl geläufigere Begriff, daher steht seine Definition an erster Stelle. Das Wort *Source* al-

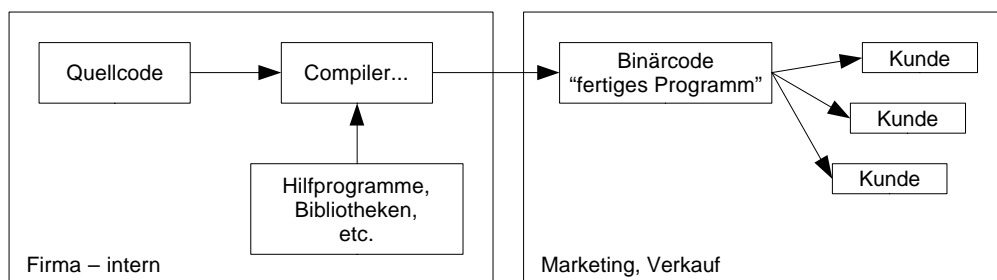


Abbildung 1: Prozeß konventioneller Softwareentwicklung: Die Sources, also der Quelltext bleibt in der Firma, nur das „fertige“ Binärprogramm wird ausgeliefert.

so *Quelle*, bezieht sich auf den sogenannten *Quelltext (Sourcecode)* von Software. Programme werden heute aufgrund der geforderten Komplexität und Flexibilität (und verschiedener anderer technischer Detailprobleme) in der Regel nicht mehr in der Sprache der Maschine also des Prozessors direkt geschrieben. In den letzten Jahrzehnten wurde eine erhebliche Anzahl sogenannter *Hochsprachen* entwickelt, also Programmiersprachen, die die gewünschte Programmlogik sowie die Benutzerschnittstellen auf abstrakterem und somit leichter handhabbarem Niveau zu entwerfen erlauben¹. Das Programm liegt also als (Hochsprachen) Quelltext vor, und wird vor der Verwendung mit Hilfe verschiedener Hilfsprogramme in das Maschinenspracheprogramm übersetzt, das sowohl der gewünschte Rechner wie auch das Ziel-Betriebssystem versteht. Dieses binäre Programm wird letztlich an den Kunden ausgeliefert.² (Dieser Vorgang ist in Abb. 1 illustriert.)

Man könnte sich—um ein etwas leichter verständlicheres Beispiel anzuführen—die Sources, also den Quellcode mit dem Bauplan eines Hauses oder den Konstruktionsplänen eines Autos vergleichen. Das eigentliche Know-How einer Software Firma liegt also in der Erstellung solcher Baupläne samt zugehöriger technischer Erwägungen und Berechnungen. Um im Beispiel zu bleiben: beim Hausbau sind vermutlich statische Berechnungen anzustellen, in der Autokonstruktion werden andere Erfahrungen erforderlich sein, wie die Kenntnisse bestimmter Materialeigenschaften usw. Aus diesem Wissen erstellen die Ingenieure die Bau oder Konstruktionspläne. Daraus kann nun (idealerweise) eindeutig das entsprechende Bauwerk oder Auto konstruiert werden; natürlich unter Zuhilfenahme anderer technischer Einrichtungen und Verfahrensweisen wie einer Baufirma, Kränen, Fertigbeton, Stahlträgern und was auch immer hier weiters erforderlich sein könnte. Dieser zweite Schritt könnte mit den genannten Hilfsmitteln der Softwarebranche wie Compiler, build und make Programmen, zugekauften Softwarebibliotheken, Entwicklungsumgebungen etc. verglichen werden. All dies interessiert den „normalen,“ Konsumenten natürlich nicht. An ihn wird einfach das Haus oder das Auto „ausgeliefert“, das in unserem Vergleich dem binären Programm entspricht, das auf der entsprechenden Hardware/Betriebssystem Kombination des Kunden lauffähig ist³.

Folglich wird der Quelltext und das Wissen zum Erstellen desselben als das Kapital einer konventionell geführten Softwarefirma verstanden und entsprechend behandelt. Dem Kunden hingegen wird diese Information nicht zugänglich gemacht. Es herrscht die Ansicht vor, das Know-How der Firma stecke genau in diesen Details, die den Quellcode der jeweiligen Applikation ausmachen.

Open Source Softwareentwicklung geht nun einen fundamental anderen Weg: Der Kern der Idee ist gerade das Offenlegen der Quellcodes, sowie klare Copyright Richtlinien die eine Veränderung und Weiterverwendung eben dieser Programme oder von Teilen daraus in der Regel nicht nur erlauben, sondern sogar als wünschenswert erachten (daher wird das Regelwerk manchmal auch *Copyleft* genannt). Open Source Programme werden in der Regel im Internet

¹Heute übliche Hochsprachen sind C/C++, Java, Basic, PHP, Perl, Python, ...

²Aufgrund der genannten Komplexität ist es allerdings selbst für Experten sehr schwierig mit dem Binärcode von modernen Software etwas anzufangen, in dem Sinne, daß er diesen für eigene Zwecke anzupassen oder weiterzuverwenden vermag. Davon abgesehen ist diese Vorgehensweise (die auch als *reverse engineering* bezeichnet wird) je nach Rechtslage teilweise sogar verboten.

³Um das Beispiel auszureizen: *reverse engineering* würde folglich dem Versuch entsprechen, aus dem fertigen Auto oder Haus das Wissen, das zur Erstellung der entsprechenden Baupläne und Berechnungen erforderlich ist zu rekonstruieren. Dies ist bis zu einem gewissen Grad wohl möglich, aber wie man sich leicht vorstellen kann, ziemlich aufwendig. Für einen Konkurrenten wäre es natürlich einfacher die Originalinformationen zur Verfügung zu haben.

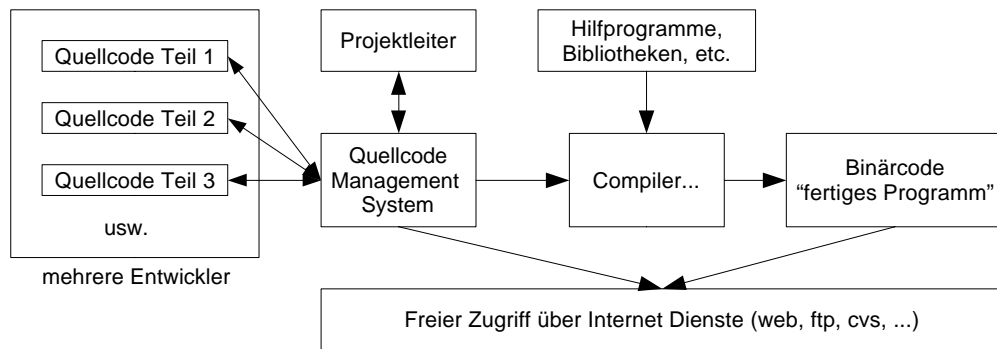


Abbildung 2: Typischer Prozeß einer Open Source Softwareentwicklung: Die Sources werden von mehreren (vielen) Entwicklern erstellt und gewartet, dann von einem (oder mehreren) Projektleitern zusammengeführt. Dieser Erstellt dann meist auch schon binäre Versionen die auch für nicht-Experten direkt verwendbar sind. Der (lesende) Zugriff auf Sourcen und Binärdaten ist für jedermann möglich.

angeboten, sowohl im Quellcode, wie meist auch in übersetzten Formen für verschiedene Betriebssysteme, um Benutzern, die weniger technisch versiert, sind die einfache Verwendung zu ermöglichen. Das Übersetzen benötigt ja Hilfsprogramme, die auch auf manchen Systemen (Windows) nicht immer automatisch zur Verfügung stehen. Ein typischer Entwicklungsprozeß wird in Abb. 2 illustriert.

Die Entwicklung der Open Source Software erfolgt zumeist ebenso über das Internet, d.h. Entwicklergruppen der ganzen Welt können sich über Internet Kommunikation (Diskussionsforen, Email, Web) koordinieren und die Entwicklung sogar komplexer Produkte erfolgreich erledigen, dazu später mehr. Zu den bekanntesten „Produkten“ zählen die Betriebssysteme BSD und Linux [18], verschiedene Internet Hilfsprogramme [11], das Apache Projekt, zu dem auch der meist verbreitete Webserver zählt [1] und der Webbrowser Mozilla, der als Nachfolgeprojekt des Netscape Browsers von der Firma Netscape in die freie Open Source Wildbahn entlassen wurde [8].

Erwähnenswert ist weiters, daß sich Projekte am Web etabliert haben, die einer große Anzahl von Open Source Projekten eine Heimat bieten, und die Entwickler mit den benötigten Quellcode Management- und Kommunikations-Hilfsmitteln versorgen und so als Dach für eine große Gruppe von Applikationen und Projekten dienen. Der Betreiber des bedeutendsten Projektes dieser Art — Sourceforge [15] — ist die Firma Redhat, die ihrerseits wiederum Geld damit verdient Linux Distributionen zusammenzustellen und dabei auf Teile der Open Source Projekte zurückgreift, die unter Sourceforge zu finden sind⁴.

Erwähnenswert ist noch die Tatsache, daß diese Begriffe (teilweise in leicht modifizierter Form wie *Shared Source*) leider in letzter Zeit fallweise auch mißbräuchlich verwendet werden. Mißbräuchlich insofern, als wohl der Eindruck erweckt werden soll, daß der eigene Ansatz die Vorteile und Eigenschaften der Open Source Applikationen integriert, dies aber tatsächlich

⁴Je nach Lizenz erlauben viele Open Source Produkte auch die Verwendung ihrer Quellcodes in kommerzieller Software, allerdings wird oft gefordert, daß die Quellcodes mitgeliefert werden. Dies ist bspw. auch bei der Redhat Linux Version der Fall.

nicht der Fall ist, weil wesentliche Eigenschaften eben nicht übernommen werden.

„The companies that have the most to gain from closed source programs and closed specifications are sometimes the quickest to misuse the terms open source and open standards⁵.“ *Ken Krechmer [6]*

2.2 Open Protocol

Der Begriff *Open Protocol* ist weniger bekannt, jedoch möglicherweise für Politik und Gesellschaft noch bedeutender als die Entwicklungen im Open Source Bereich. Ein Protokoll ist eine technische Sprache, die es verschiedenen Systemen erlaubt miteinander zu kommunizieren. Eines der bekanntesten Protokolle ist das *hypertext transfer protocol* (http). Dieses beschreibt die Art und Weise in der Webbrowser und Webserver (also das Programm, welches die Daten, z.B. Webseiten und Bilder ausliefert) miteinander kommunizieren. Um Mißverständnissen vorzubeugen: http ist vergleichbar mit einem adressierten Briefkuvert und einer Briefmarke, die Kommunikation zwischen Webserver und Webbrowser mit der Post, die den genannten Brief von Sender zu Empfänger übermittelt, aber dies eben nur unter der Voraussetzung, das sowohl Brief, als auch Marke eine bestimmte definierte Form haben. Über den transportierten *Inhalt* sagt das Protokoll (http als Beispiel) jedoch nichts aus. Nur über die *Form* des Inhaltes. Ein frankiertes Briefcuvert ist beispielsweise geeignet Papier mit textueller Information zu transportieren, nicht aber eine Blumenvase oder Weinflaschen. Dafür wäre ein anderes Protokoll, nämlich das „Paket“ geeignet.

Protokolle können also als die Basis-Kommunikationsinfrastruktur unserer modernen Informationstechnologie (IT) Welt verstanden werden, und sind folglich ubiquitär anzutreffen: in Telephon, Mobilkommunikation, World Wide Web, EDI (elektronischer Datenaustausch zwischen Firmen) usw. Entsprechend groß ist folglich die Bedeutung von Protokollen für das tägliche Leben, das ja immer mehr von technischer Infrastruktur durchdrungen ist⁶.

Ähnlich dem vorigen Abschnitt, gibt es nun *offene* und *geschlossene/proprietäre* Protokolle. Proprietäre Protokolle sind solche, die von einer Firma definiert werden und deren Verwendung für andere entweder gar nicht, oder nur unter bestimmten Auflagen möglich ist. Große Firmen versuchen seit langem Marktdominanz und Monopole mit solchen Hilfsmitteln zu erhalten oder zu festigen. Als Beispiele sei der HTML Standard genannt, der vor einigen Jahren von Microsoft und Netscape einigermaßen willkürlich verändert und erweitert wurde, nur um die Nutzer an den jeweils eigenen Browser zu binden, oder auch die bewußt schlechte Dokumentation der Microsoft Office Formate, sowie deren dauernde Veränderung mit jedem Versionswechsel, die im wesentlichen dazu dient, Kunden dazu zu zwingen jeweils die neueste Version des Produktes zu kaufen. Selbst wenn man mit einer älteren Version zufrieden wäre, bzw. Nutzer eines Konkurrenzproduktes ist, gerät man so unter Druck, da man schließlich Dokumente der neuesten Version bekommt, und mit diesen Arbeiten muß, was eben mit den Vorgängerversionen dann nicht mehr möglich ist.

⁵Die Firmen, die am meisten an Closed Source und verschleierte Spezifikationen zu gewinnen haben, sind manchmal die Ersten, die die Begriffe *Open Source* und *Open Standard* mißbrauchen

⁶Manchmal werden auch Dateiformate wie die *hypertext markup language (HTML)* — dabei handelt es sich um die Auszeichnungssprache die dazu dient Informationen am World Wide Web zu publizieren — oder Daten von Office Produkten (Textverarbeitung, Tabellenkalkulation, ...) in diesem Umfeld erwähnt, auch wenn sie nicht der klassischen Definition von Protokollen genügen.

Diese Beispiele zeigt, wie Monopolisten ihre Stellung auf dem Rücken der Konsumenten ausnutzen und versuchen unter Zuhilfenahme (man könnte oft sogar sagen: unter Mißbrauch) von Protokollen und Datenformaten und nicht durch Qualität oder Funktionalität ihrer Software Kunden zu binden, ja ich bin sogar geneigt zu sagen: in manchen Fällen sogar in Geiselschaft zu nehmen. Offene Protokolle hingegen sind Versuche solche wesentlichen Kommunikationswege in einer offenen und einigermaßen demokratischen Weise zu entwickeln und zu beschreiben, da diese Spezifikationen letztlich frei verfügbar allen Programmierern zu Verfügung stehen. Entweder in gedruckter Form oder heute eher in elektronischer Form. Und nicht nur das: auch die Teilnahme am Entwicklungsprozeß steht grundsätzlich allen Interessierten Gruppen oder ggfs. auch Einzelpersonen offen. Als Beispiel sei das World Wide Web Konsortium (W3C) genannt [19]. Dieses Konsortium umfasst alle großen Softwareproduzenten, sowie Vertreter staatlicher wie nicht-staatlicher Organisationen, Universitäten etc. Im Rahmen dieses Konsortiums werden Protokolle und Formate spezifiziert die im Rahmen des World Wide Webs, sowie des Datenaustausches von hoher Relevanz sind wie: HTML, XML, RDF, SVG ...

Aus genannten Gründen ist folglich größtes Augenmerk darauf zu legen, daß in allen relevanten Bereichen nur offen standardisierte Protokolle verwendung finden. Dies ist auch ein klarer Auftrag an gesetzgebende Instanzen und garantiert einerseits Transparenz der ablaufenden Vorgänge⁷ sowie die Möglichkeit der Konkurrenz unter Software Anbietern, da sich jede Firma im Prinzip mit ihren Produkten in offene Schnittstellen und Protokolle „einklinken“ kann, was durch die Verwendung proprietärer „Standards“ erheblich erschwert, wenn nicht unmöglich wird. Eine derartige Standardisierung ist letztlich der einzige Garant für gesunde Vielfalt im Bereich der IT Infrastruktur. Allerdings muß dies natürlich in vielen Fällen auch gegen starke Lobbies der jeweiligen Monopolisten durchgesetzt werden.

3 Unterschiede zwischen „offenen“ und konventionellen Entwicklungsprozessen

Es ist klar, daß sowohl der Entwicklungs- als auch der Geschäftsprozess sich zwischen offenen und konventionellen Prozessen deutlich unterscheiden. Ist der „konventionelle“ Entwicklungsprozeß von Software mittlerweile eine eigene wissenschaftliche Disziplin (*Software Engineering*) und wird letztlich auch der Versuch unternommen dies als Ingenieurs-Disziplin bspw. im Rahmen des Informatik Studiums zu definieren, so hat es alle Beteiligten verwundert, daß es überhaupt möglich ist, komplexe Produkte ohne solche klaren strategischen Prozesse zu entwickeln. Die Open-Source Szene hat gezeigt, daß dies in verschiedenen Bereichen sehr wohl möglich ist, und ohne auf Details einzugehen [7, 21] die Qualität der Produkte sogar vielfach die Qualität konventioneller kommerzieller Produkte übersteigt (Apache Webserver, Linux, ...). Dies wird unter anderem durch die extensive Nutzung des Internets als Kommunikations-Medium sowie den offenen Quellcode, sowie die offene Diskussion über Features und Fehler(!)

⁷Es sei auch an sicherheitsrelevante Bereiche wie staatliche Verwaltung, Gesundheitssystem etc. gedacht. Bei offenen Protokollen ist für den *internen* Experten jederzeit einsichtig welche Daten wann und aus welchem Grund zwischen welchen Systemen übertragen werden. Bei proprietären Protokollen hingegen nicht! Es wundert mich sehr mit welchem Vertrauen staatliche Organisationen irgendwelchen Firmen vertrauen, welche Daten diese zwischen Ihren Applikationen und im Zeitalter des Internets möglicherweise sogar zur eigenen Firma übertragen. Erwähnt seien nur kurz die Versuche des amerikanischen Geheimdienstes Hintertüren in Verschlüsselungssysteme amerikanischer Softwareprodukte einzubauen. **Selbstverständlich** werden solche Hintertüren **niemals** bspw. für Industriespionage genutzt!!

ermöglicht⁸.

Allerdings ist ein wesentlicher Unterschied sehr wohl zu bemerken: Während kommerzielle Software immer ein bestimmtes Ziel vor Augen hat (Zeitplan, Funktionen, Marktpositionierung) ist dies bei Open Source Software in der Regel nicht der Fall. Oft entstehen OS Projekte aus konkreten Problemen einzelner Programmierer (die nicht-Verfügbarkeit eines lizenzfreien Unix-artigen Betriebssystems führte durch Linus Torvalds zur Initiierung des Linux Projektes [18]). Wenn diese Arbeit aber auf breiteres Interesse stößt, entwickelt sich dann eben in manchen Fällen ein größeres Projekt, das oftmals auch durch Unterstützung der Industrie vorangetrieben wird. Ein Beispiel hierfür wäre die Firma IBM, die in den letzten Jahren verschiedene OS Projekte finanziell unterstützt [1], bzw. eigene Entwicklungen für die OS Szene öffnet, zu nennen.

Ein interessanter Aspekt von Open Source Software ist auch die Frage nach Unterstützung (Support) im Fall von Problemen, Fehlern (Sicherheitslücken) oder nötigen Anpassungen. Neben der später beschriebenen Tatsache, daß mittlerweile viele Firmen in diesem Bereich gute Geschäfte machen, war es immer eine erfreuliche Tatsache vieler Projekte, daß der Support oftmals eine bessere Qualität hat als bei so manchen kommerziellen Produkten. Der Grund ist folgender: Da sich OS Projekte geradezu zwangsläufig in der Öffentlichkeit abspielen, sind auch die Entwickler auf das Feedback der Benutzer angewiesen; außerdem ist es für interessierte Experten durch die Offenheit der Quelltexte verhältnismäßig leicht, sich Überblick über die Details der Systeme zu verschaffen. Die notwendige Kommunikation zwischen Entwicklern und Nutzern erfolgt in der Regel über Internet Diskussionsforen oder Mailing-Listen.

Die Konsequenz ist, daß man bei Fragen oder Problemen einerseits üblicherweise bereits nach Stunden qualitativ hochwertige (und kostenlose!) Antworten erhält, andererseits aber dieses Feedback den Entwicklern wesentliche Anregungen für Verbesserung und Fehlerkorrektur liefert. Der Prozess ist hier also von vielschichtigeren und kürzeren Etappen geprägt, als in konventionellen Entwicklungsprozessen mit „geheimen“ Quelltexten. Die Konsequenz ist folglich nicht, daß Software sofort über eine höhere Sicherheit oder Qualität verfügt als kommerzielle Software, aber sobald das Produkt eine gewissen Zeit in Verwendung ist, werden Fehler und potentielle Sicherheitslücken schneller gefunden und korrigiert als oftmals in kommerziellen Systemen. Der Grund liegt natürlich auch im „philosophischen“ Ansatz des Prozesses: Eine Firma hat naturgemäß kein übermäßiges Interesse mit Fehlern oder Sicherheitslücken in der Öffentlichkeit konfrontiert zu werden, die Open Source Szene lebt aber geradezu von der öffentlichen Diskussion. Dies führt dazu, daß bspw. der bekannteste Open Source Browser Mozilla nach der Ansicht der meisten Experten als wesentlich sicherer anzusehen ist, als das Konkurrenzprodukt des Marktführers [13, 12]. Die Gründe dafür werden auch im nächsten Kapitel ausführlicher diskutiert.

Dies hat sogar Bill Gates mittlerweile erkannt und das sogenannte *trustworthy computing* also die Sicherheit des Systems vor allen neuen Funktionen der Programme als vorrangig erklärt. Allerdings bleibt bei vielen die Frage offen, ob es sich um eine tatsächlich neue Orientierung des Konzernes handelt, oder wieder einmal nur um geschicktes Marketing. Letztlich hätte dies weitreichende Konsequenzen, die die gesamte Microsoft Strategie der nächsten Jahre nachhaltig verändern, müsste wie folgendes Zitat erläutert:

⁸Der Schluß, der aus diesen Erfolgen für den Wissenschaftszweig des Software-Engineering zu ziehen wäre, geht natürlich weit über diesen Text hinaus und es bleibt den Experten der Disziplin ihre Schlüsse daraus zu ziehen. Tatsächlich sind diese Fragen ja auch Teil der aktuellen Forschung wie beispielhaft folgende Artikel zeigen: [7, 3].

„Bill Gates is correct in stating that the entire industry needs to focus on achieving trustworthy computing. He’s right when he says that it is a difficult and long-term challenge, and I hope he’s right when he says that Microsoft is committed to that challenge. I don’t know for sure, though. I can’t tell if the Gates memo represents a real change in Microsoft, or just another marketing tactic. Microsoft has made so many empty claims about their security processes – and the security of their processes – that when I hear another one I can’t help believing it’s more of the same flim-flam. [...]

And they’re going to have to reverse their mentality of treating security problems as public-relations problems. I’d like to see honesty from Microsoft about their security problems. No more pretending that problems aren’t real if they’re not accompanied by exploit code, and attacking the security researcher if they are. No more pretending security problems aren’t caused by bad code in the first place. No more claiming that XP is the most secure operating system ever, simply because it’s the one they want to sell.“ *Bruce Scheider [12]*

Es bleibt also die Frage offen, ob das Engagement nur als weiterer Marketing Gag zu betrachten ist, oder ob es sich um eine strukturelle Neuorientierung des Konzernes handelt. Letzteres wäre zwar aus der Sicht der Konsumenten wünschenswert, ist aber nicht als sehr wahrscheinlich zu betrachten, weil es so grundsätzliche Änderungen erfordern würde (wie im Zitat beschrieben wird), daß es die Produktneuentwicklung der nächsten Jahre massiv beeinträchtigen würde. Es scheint vielmehr eine komplett neue Strategie geplant zu sein, die die Sicherheit nach außen verlagert, indem nicht die Systeme in sich sicher gemacht werden, sondern die installierten Systeme sowie deren Benutzung beim Benutzer selbst überwacht werden sollen (TCPA, Palladium). Dies soll an anderer Stelle ausführlicher diskutiert werden.

4 Alternativen und Risiken

Der im vorigen Abschnitt erwähnte grundsätzliche Unterschied im Zugang und im Entwicklungsprozeß hat natürlich noch weitere Konsequenzen, bzw. wirft Fragen auf. Zwar ist es in der Regel möglich auf Probleme bspw. mit Linux oder Apache Projekten innerhalb von Stunden oder Tagen qualitativ hochwertige Lösungen zu bekommen, allerdings ist dies eben nicht garantiert. Das ist eine klare Konsequenz aus der Tatsache, daß es sich um freiwillige Gemeinschaften handelt, wo eben eine Verpflichtung nicht gegeben ist. Diese Tatsache ist für viele Bereiche unproblematisch, da man eben in praktisch allen Fällen zu einer Lösung findet, für manche Geschäfts- oder Anwendungsfälle ist dies aber nicht akzeptabel. Man denke sich eine Bankenlösung die auf Open Source Software basiert, wo aber garantierte Antwortzeiten im Supportbereich einfach erforderlich sind. Hier wird man sich nicht auf freiwillige Internet-Communities verlassen können und wollen, sondern vielmehr kommerziellen und garantierten Support wünschen. Dieser ist mittlerweile auch verfügbar. Sowohl von Spezialisten wie bspw. Suse oder Redhat im Linux, SAP im Business Software Bereich oder von Komplettanbietern wie IBM, Sun oder HP. Somit greift auch das Argument mangelndes Supportes längst nicht mehr.

Dennoch ist dieser Punkt natürlich zu bedenken, wenn man plant entsprechende Software einzusetzen. Nicht für jedes Open Source Produkt ist guter kommerzieller Support verfügbar.

Allerdings ist der Unterschied zu kommerzieller Software nicht gar so groß, wie ihn manche Firmen gerne darstellen wollen, denn die Supportqualität im kommerziellen Bereich ist kostenlos mittlerweile nicht mehr vorhanden, und der bezahlte Support in manchen Fällen auch kaum verfügbar⁹. Außerdem besteht bei Closed Source (CS) Software immer die Einschränkung, daß man das Programm an sich nicht verändern kann, z.B. um Anpassungen an eigene Probleme durchzuführen, oder nach Sicherheitslücken zu suchen, oder diese schnell zu korrigieren. Dies ist aber gerade in sicherheitskritischen Bereichen eine wesentliche Sache, wie das Beispiel der Bankensoftware illustrieren kann. Im Falle von CS Software kann gar nicht aktiv überprüft werden, ob an bestimmten Stellen Probleme zu erwarten sind¹⁰, und wenn in der Praxis kritische Fehler auftreten, muß der Hersteller kontaktiert und eine Korrektur veranlasst werden. Ist OS Software im Einsatz ist es oftmals möglich solche Korrekturen unmittelbar selbst anzubringen und somit kritische Fehler wesentlich besser und schneller zu beheben.

Ein weiterer manchmal geäußerter Kritikpunkt besteht in der Tatsache, daß Weiterentwicklung von OS Produkten nicht garantiert ist, weil sie ja von Freiwilligen getragen wird, und somit diese sozusagen nach Lust und Laune die Entwicklung einstellen können. Diese Argumentation erweist sich sehr schnell als Bumerang. Denn erstens ist dies auch im kommerziellen Umfeld nicht garantiert — es ließen sich zahlreiche Beispiele für Produkte, deren Entwicklung eingestellt wurde finden¹¹ — und andererseits ist dieses Problem genau genommen im Closed Source Bereich noch wesentlich problematischer als im Open Source Bereich. Der Grund ist folgender: Wird ein CS Produkt eingestellt, weil der Hersteller dieses nicht mehr unterstützen will, oder in Konkurs geht, so haben die Kunden ein kaum lösbares Problem, weil die Programmquelltexte ja nicht frei sind, und eine Weiterentwicklung schon prinzipiell nicht möglich ist. Wird ein OS Projekt nicht mehr weiterentwickelt, so ist dies zwar auch problematisch, aber in der Praxis finden sich bei guten Projekten in der Regel neue Teams, die die Entwicklung weiterführen und falls dies auch nicht der Fall sein sollte, so besteht immerhin die Möglichkeit das Produkt selbst weiter zu pflegen. Mir ist ein konkreter Fall bekannt, wo dies im Rahmen der Softwareentwicklung im Versicherungssektor passiert ist. Ein Applikationsserver wurde nicht mehr weiterentwickelt, da es sich aber um eine wesentliche Komponente der Versicherungssoftware gehandelt hat, wurde die Pflege und Entwicklung vom Versicherungsteam selbst weitergeführt (jedenfalls die Leitung des OS Projektes).

Es zeigt sich auch, daß es von erheblichem Vorteil sein kann, wenn bei größeren Projekten die zu Grunde liegende Infrastruktur eben keine CS *black-box*¹² ist, sondern vom Entwickler-

⁹Dies mag unter anderem auch daran liegen, daß Firmen heute prinzipiell aufgrund der neuen Vertriebskanäle (Internet...) leicht weltweit operieren können, dies jedoch für den Support — der ja in schwierigen Fällen, wo schnelle persönliche Interaktion gefordert ist, nicht ohne großen Aufwand globalisierbar ist, da ja lokale Strukturen auf allen Märkten etabliert werden müssen — nicht ohne weiteres gilt.

¹⁰Außerdem ist Transparenz ein wesentliches Kriterium. Schließlich ist es wünschenswert zu verstehen, was genau mit den geschäftskritischen Daten passiert. Black-Box Verfahren zeigen hier wesentlich mehr Unsicherheiten (z.B. akzeptieren im Bereich der Kryptographie Experten keinesfalls Verfahren, deren genaue Funktionsweise und Quelltext nicht verfügbar ist!).

¹¹Man denke an den hochinteressanten Tabellenkalkulations-Ansatz Lotus Improv, für die nichteinmal Importfilter in modernen Tabellenkalkulationen zur Verfügung stehen!

¹²Unter einem black-box System versteht man ein solches, das bestimmte Funktionalität nach außen hin in Form von Schnittstellen anbietet, und aufgrund bestimmter Eingaben dann bestimmte Dinge tut, oder Ergebnisse liefert. Das entscheidende ist jedoch (daher black), daß der Benutzer darüber im unklaren gelassen wird, wie genau die Ausgaben oder Ergebnisse zustande kommen. Dies bleibt ein Geheimnis. Man muß folglich darauf vertrauen, daß dieses System korrekt arbeitet, hat aber keine Möglichkeit die Arbeitsweise nachzuvollziehen oder zu überprüfen.

team verstanden wird. Das ganze System ist so leichter handhabbar und wartbar¹³. Besonders in großen Projekten für Banken, Versicherungen, Krankenanstalten, Verwaltung etc. ist die Kapazität oft vorhanden solche Projekte zu pflegen und das Risiko im Falle von OS Software somit dramatisch niedriger als im Falle von CS Produkten.

Gerade die genannten Aspekte sind ein wesentlicher Aspekt der freien Meinungsäußerung und Demokratisierung einer Gesellschaft und deren Wirtschaft. Die Tatsache, daß sich prinzipiell jeder an der Entwicklung eines OS Projektes beteiligen kann, ist als wesentlicher Fortschritt und vielleicht eine der größten Errungenschaften des Internets in diesem Bereich zu verstehen. Und sollte er sich mit den Zielen des Projektes nicht oder nicht mehr einverstanden erklären, so hat er jederzeit die Möglichkeit ein eigenes neues Projekt auf die Beine zu stellen und die Quelltexte des bisherigen als Grundlage zu nehmen. Auch reale Beispiele für diesen Fall existieren (Interbase/Firebird Datenbankentwicklung, PHPNuke/PostNuke Content Management System ...).

5 Wirtschaftliche Konsequenzen

“Cooperation is more important than copyright. But underground, closet cooperation does not make for a good society. A person should aspire to live an upright life openly with pride, and this means saying *No* to proprietary software. [...] You deserve free software”, Richard Stallman

Firmen wie IBM, SGI, Sun, HP, Redhat, O'Reilly [9] etc. zeigen daß Open Source Software und kommerzieller Erfolg nicht in einem Gegensatz stehen müssen. Immer komplexer werdende Systeme und deren Implementation, Wartung und Schulung beim Kunden erzeugen ein erhöhtes Maß an benötigter Unterstützung, die diesen Firmen wiederum Gewinn garantiert. Außerdem kommen oftmals Synergien zu tragen: Wenn IBM Linux auf seiner gesamten Serverpalette vom z-Series Großrechner bis zum Desktop unterstützt, so geschieht dies klarerweise auch nicht aus reiner „Selbstlosigkeit“, sondern in der (durchaus berechtigten) Hoffnung, mit stabilen und offenen Systemen auch gute Geschäft mit der nötigen Hardware, sowie Beratung und Schulung zu machen.

Die andere Seite ist jedoch die Frage, wie sich die Entwicklung von Open Source Software finanziert. Diese Frage ist weniger leicht pauschal zu beantworten. Teilweise geschieht dies durch genannte Firmen, die manche eigenen Projekte „öffnet“. Viele Projekte entstehen letztlich durch staatliche Subventionen, sei es direkt in Form von Projektgeldern oder indirekt in Form von Entwicklungen die von Universitäten getragen werden. Einige Projekte werden auch tatsächlich von Hobby-Programmierern geführt, die diese Systeme in ihrer Freizeit entwickeln. Es zeigt sich, daß es sich auch für große Firmen wie IBM oder SAP auszahlen kann, manche Produkte, im OS Umfeld zu positionieren, zumal sich dann auch eine weltweite Community etablieren kann, die Teile der Entwicklung zu übernehmen vermag, sowie die typische OS Interaktion begleitet und die Qualität der Produkte stark vorantreibt¹⁴.

¹³Nicht umsonst fordern manche große Firmen bzw. staatliche Organisationen auch bei konventionellen Produkten Quelltext Einsicht.

¹⁴Als Beispiele könnte man das relationale Datenbankmanagement der Firma SAP nennen: SAPDB, einige XML Projekte im Apache Pool, die von IBM „gespendet“ wurden, oder auch Journaling Filesysteme von IBM oder SGI.

Auch ein interessanter weiterer Aspekt ist zu sehen: Es ist zu beobachten, daß die Teilnahme, bzw. das Finanzieren von Open Source Projekten sich sehr günstig auf das Image der Firma auswirkt. Dies konnte in den letzten Jahren stark bei IBM beobachtet werden, aber auch bei anderen Firmen wie der genannten SAP. Auch Sun ist hier verstärkt bemüht, wenn man die Unterstützung der Apache Projekte bzw. von Netbeans verfolgt. Allerdings sind die Vertreter der Open Source Szene hier sehr kritisch und eine Unterstützung, die sich als reines Marketing erweist, wirkt stark kontraproduktiv! Dies ist auch der Grund, warum die *Shared Source* Strategie von Microsoft von Anfang an zum Scheitern verurteilt war: Es war unmittelbar klar, daß der Gedanke dahintersteht die Kapazitäten der *community* zur Verbesserung der eigenen Applikationen zu nutzen, ohne einen adäquaten Gegenwert zurückzugeben.

6 Bezüge zu Politik und Gesellschaft

„It [Europe] talks about catching up but knows that the technological gap between it and the United States grows larger every day. The continent that invented culture now imports its culture from America. The equivalent of “Intel Inside” could be printed on almost everything new in Europe.“ *Lester C. Thurow [17]*

Noch im Bezug auf die wirtschaftlichen Konsequenzen ist zu bemerken, daß einige Dinosaurier der Branche (die allerdings noch über eine erhebliche Kriegskasse verfügen) sich nach wie vor auf Geheimhaltung ihrer Quellcodes beschränken, sowie den Versuch mit proprietären Formaten und Protokollen den Markt zu behindern. Dieses *Geschäft mit dem Geheimnis* geschieht zum Schaden des Konsumenten und der Mitbewerber. Das gelingt darum so hervorragend, weil es die Politik (auch, oder gerade auch die europäische) bisher verschlafen hat, entsprechende regulative gesetzliche Rahmenbedingungen zu schaffen um solche schädlichen Monopole zu verhindern. Im Gegenteil stehen mit den neuen Ansätzen im Bereich der Softwarepatente, des angeblichen Schutzes von Urheberrechten sowie dem neuerlichen Übersehen der Konsequenzen von Entwicklungen wie Palladium und TCPA neue noch viel größere politische Fehler vor der Tür.

Auch sollte die allgemeine gesellschaftspolitische Dimension des Problem es nicht unterschätzt werden. Wir bewegen uns auf eine Gesellschaft zu, in der die wesentlichen Bereiche von sehr wenigen großen Firmen kontrolliert werden. Sei dies im Bereich der Desktop Computer, der Buchhaltungs-Software, der Mobilkommunikation, oder weitergehend in Bereichen wie E-Government. Gerade letzteres *Buzzword* demonstriert die Problematik! Ich zweifle daran, daß es wünschenswert sein kann, wenn ein großer weltweit agierender Monopolist in wenigen Jahren möglicherweise den Großteil der Staaten mit *proprietärer* (!) Software für Verwaltung und demokratische Prozesse ausrüstet. Nicht nur, daß dies aus demokratiepolitischer Sicht wegen mangelnder Transparenz kaum wünschenswert sein kann, ist dies auch finanziell hochgradig problematisch, weil man sich letztlich freiwillig in die Abhängigkeit begibt.

Ein weiterer noch nicht erwähnte Faktor ist die Lage der sogenannten *zweite Welt Staaten* oder *Schwellenländer*. Genannt seien exemplarisch China, Indien, Indonesien, Iran, Korea, Brasilien usw. Diese Länder scheinen mit Recht zu erkennen (viel weitsichtiger offenbar als die Staaten der europäischen Union), daß OS Software für sie eine wesentliche Komponente des Aufbaues eigener Infrastruktur sein kann und sein wird, da sie einerseits eine kostengünstige Möglichkeit darstellt, andererseits aber auch erlaubt Know-How zu erwerben und weiterzuentwickeln. Im Gegensatz zum Einsatz von proprietären Produkten, die eine rein „passive“ Ver-

wendung erlauben. Obwohl die bekannten Monopolisten versuchen in diese Märkte mit kurzfristig äußerst günstigen Konditionen einzudringen, scheinen die genannten Staaten offenbar zu erkennen, daß sich dies zu ihrem Nachteil entwickeln würde. Sind die (amerikanischen) Monopolisten erst einmal Marktführer in China, Indien, Brasilien etc, und hängt ein wesentlicher Bereich der Infrastruktur an deren Produkten, so wird mit Sicherheit die Preisschraube massiv angezogen und diese Staaten würden in Zukunft, obwohl ohnedies in finanziellen Schwierigkeiten, in noch größere Abhängigkeiten schlittern.

Während es der europäischen Union offenbar noch nicht klargeworden ist, wie groß und v.a. wie unnötig die Abhängigkeiten geworden sind, so scheinen die Schwellenländer hier wesentlich aufmerksamer zu sein und ihre Infrastruktur selbst in die Hand nehmen zu wollen¹⁵. Diese Politik befreit die Entwicklungsländer einerseits von globalen Abhängigkeiten und fördert andererseits globale offene Systeme und Standards zum Nutzen aller und nicht nur einzelner Monopolisten oder Staaten. Bedenkt man die Summen, die bspw. die EU jedes Jahr in Form von „Microsoft-“ oder „Oracle-“ etc. „Steuer“ freiwillig in die USA abführt, und die Tatsache, daß Linux und andere große Open Source Projekte sehr stark europäisch getragen werden, so offenbart sich die politische und ökonomische Kurzsichtigkeit. Würde ein Teil dieser Mittel zur Förderung der OS Communities aufgebracht werden, könnten Produkte genannter Firmen und auch andere ohne weiteres durch offene, europäische Produkte ersetzt werden. Weiters würde IT Know-How und Forschung (die Fundamente zukünftiger ökonomischer Erfolge!) im EU Raum verstärkt werden; und ein implizite Förderung schwächerer Länder wie der genannte Schwellenländer wäre erzielbar — ein höherer Grad an Transparenz und Demokratie ohnedies! Es scheint als wäre Deutschland eines der wenigen Länder in der europäischen Union das dies wenigstens ansatzweise erkannt hat und entsprechende erste kleine Schritte setzt (Explizite Förderung von OS Projekten durch Ministerien etc.). Als konkreteres Beispiel könnte die Strategie dienen, alle Server des deutschen Bundestages von Windows auf Linux umzustellen [2] sowie die Entscheidung von Schwäbisch Hall nicht nur Server sondern auch die Clients, also die Computer der Endbenutzer auf (SuSe) Linux umzustellen [14]. Interessant sind einige Zitate aus der genannten Pressemitteilung:

„Die Ankündigung der Firma Microsoft, ihr Betriebssystem Windows NT 4.0 ab dem Jahr 2003 nicht mehr zu unterstützen, zwingt den Deutschen Bundestag, für seine ca. 5000 Arbeitsplatz- und Serversysteme eine Migration zu einem Nachfolgesystem vorzunehmen.

Mit der im Quellcode offenliegenden und lizenzgebührenfreien Open Source Software existieren heute Alternativen zu den Microsoftprodukten. Damit besteht auch die Möglichkeit, *die Abhängigkeit von einem Unternehmen zu reduzieren.*

[...]

Eine solche Entscheidung bedarf umfangreicher Untersuchungen.

[...]

[die] Kommission [ist] zu folgenden Empfehlungen gelangt:

¹⁵Manche staatliche Organisationen begeben sich sogar freiwillig in Abhängigkeiten. Während sogar in den USA staatliche Internet Dienste ohne Zugangshindernisse ausgestattet werden müssen (bspw. behindertengerecht), steht auf der Homepage des österreichischen Finanzministeriums: „Optimiert für Internet Explorer. ...“

- Die Server werden auf das Betriebssystem Linux umgestellt, als Verzeichnisdienst wird OpenLdap eingesetzt.
- Auf den Arbeitsplatzrechnern wird Microsoft XP und das entsprechende Office Paket eingesetzt.
- Als Standard für Browser und eMail Client wird weiterhin Netscape genutzt.

[...]

In der Studie war die Variante, bei der nur einige Server auf Linux, der große Rest der Server aber auf Windows umgestellt worden wäre, bei der Nutzwertanalyse auf dem ersten Platz gelandet. Entsprechend wäre als Verzeichnisdienst Active Directory eingesetzt worden.

Die Variante, der die Kommission jetzt zugestimmt hat, hatte bei der Nutzwertanalyse den zweiten Platz erreicht.

Mit dieser Entscheidung weicht die IuK- Kommission bewusst vom Ergebnis der Studie ab, *weil sie mit dieser Entscheidung die strategische Überlegung verbindet, die zur Zeit bestehende Abhängigkeit von den Produkten eines Anbieters zu lockern. Sie erhofft sich durch diesen Schritt einen größeren Freiraum bei zukünftigen Entscheidungen.*“
Pressemitteilung Deutscher Bundestag (Hervorhebung vom Autor) [2]

Diese Entscheidung sowie deren Begründung ist offensichtlich eine weitsichtige und berücksichtigt nicht nur kurzfristige finanzielle Aspekte. Es wurde sehr klar erkannt und formuliert, daß die Entscheidung für ein offenes System auch die von mir geschilderten Vorteile hat, daß nämlich in Zukunft nicht nur ein Anbieter (oder Business Partner dieses Anbieters) zum Zug kommen können, sondern vielmehr eine breite Palette auch an inländischen Klein- und Mittelbetrieben. Das Risiko wird durch diese offene Variante deutlich minimiert, die Transparenz erhöht, sowie der Wettbewerb gestärkt. Diese Entscheidung ist natürlich nur ein kleiner, aber immerhin ein wichtiger erster Schritt in die richtige Richtung.

Ähnlich die Aussagen in der Pressemitteilung der Stadt Schwäbisch Hall:

„Die Stadtverwaltung Schwäbisch Hall hat sich für den Aufbau einer vollständig linux-basierten IT-Infrastruktur entschieden. Durch den Einsatz von SuSE Linux auf Servern und Desktop-PCs wird eine vorhandene Windows-Infrastruktur abgelöst. Das Einsparpotential durch diese Lösung liegt im sechsstelligen Euro-Bereich und trägt maßgeblich zur Entlastung des Haushalts der 36.000 Einwohner zählenden Kommune bei.

[...]

[Der] Oberbürgermeister [...] erläutert die Entscheidung der Stadtverwaltung: “Für mich gibt es drei Gründe, um auf Linux zu setzen. Erstens erwarte ich eine deutliche Kostenreduktion unserer Ausgaben im IT-Bereich durch die Senkung der Software-Lizenz-Gebühren. [...] Zweitens soll unsere IT-Struktur sicherer werden; die Fachleute stellen Linux in dieser Hinsicht hervorragende Noten aus. *Drittens setzen wir auf offene Standards, die eine Wahlfreiheit bei den technischen Angeboten sicher stellen.*”

[...]

Eingebunden in das Projekt ist auch ein umfangreiches Servicekonzept, das die Eröffnung eines Linux-Kompetenz-Centers in der Stadtverwaltung vorsieht. Dieses dient in der ersten Ausbaustufe zur Schulung und Information der Mitarbeiter, in weiteren Schritten auch zur Nutzung durch örtliche Schulen, Krankenhäuser und die Öffentlichkeit, um die Potentiale von Linux-Lösungen möglichst weiten Teilen zugänglich zu machen. Darüber hinaus sind auch der Aufbau einer Hotline für die Mitarbeiter, zielgruppengenaue Schulungen zu Linux und Beratungsangebote für die Auswahl und Umsetzung von weiteren linux-basierten Anwendungen vorgesehen.“ Pressemitteilung Schwäbisch Hall (Hervorhebung vom Autor) [14]

Die Entscheidung der Stadtverwaltung von Schwäbisch Hall ist aus verschiedenen Gründen als noch interessanter zu bezeichnen: Zunächst beschränkt sich die Stadt Schwäbisch Hall nicht alleine auf die Modernisierung der Serverseite, sondern schließt explizit auch die PCs der Anwender mit ein. Dieses Beispiel zeigt, daß dies heute in vielen Anwendungsbereichen nicht nur ohne nennenswerte Einschränkungen, sondern vielmehr mit Gewinn an Flexibilität, Sicherheit sowie unter Kostenersparnis möglich ist. Weiters wurde hier auch sehr weitsichtig erkannt, daß die Nutzung offener Standards auch den Vorteil bringt, daß sich die Stadt nicht auf lange Zeit an einen einzigen Partner bindet, sondern daß verschiedene Anbieter aufgrund der Transparenz des Systemes Lösungen anbieten und integrieren können. Eine weitere sehr wichtige Erkenntnis wurde im Zuge eines Gesamtkonzeptes auch gemacht: Die Gründung eines Linux-Kompetenzzentrums ist Teil der Umstellung. Ein solcher Schritt sollte es allen Interessierten im öffentlichen Sektor sowie auch in der Industrie leichter machen auf offene transparente Systeme umzustellen, da hier Erfahrungsaustausch sicherlich noch notwendig ist.

Weitere Schritte — idealerweise der gesamten EU — wären dringend erforderlich im besonderen in Folge drohender Gefahren wie TCPA [16] und Palladium auf *digitale Freiheit* und Selbstbestimmung! Es erschiene beispielsweise sinnvoll, ein Konzept der Open Source Kompetenz-Zentren nicht nur auf lokale Initiativen beruhen zu lassen, sondern vielmehr in einer EU-akkordierten und geförderten Aktion zu begründen.

Es geht auch längst nicht mehr um die Frage, ob ein bestimmtes Produkt eines Monopolisten besser oder schlechter als dasjenige eines Konkurrenten ist; oder ob es wenigstens ausreichend für die gestellten Aufgaben ist. Selbst wenn das Produkt ausgezeichnet sein sollte, so ist es ein indiskutabler Zustand, wenn eine Firma mit einem closed-source Produkt einen gesamten Markt dominiert. Dies ist eine äußerst gefährliche Entwicklung für die staatliche und ökonomische Souveränität. Erwägt man nun als Alternative (bspw. als europäischen Ansatz) die massive Subvention von Open Source Produkten, die dann in der gesamten europäischen Union eingesetzt werden können, so hat dies schon mittelfristig einen enormen Kostenvorteil, sowie auch die Klarheit und Transparenz, die man sich in solchen kritischen Bereichen wünscht.

7 Dystopische Entwicklungen?

7.1 Dystopien

Verwendet man das Wort *Dystopia* im Titel, sollte man wohl erläutern was darunter zu verstehen ist. Tatsächlich ist dies aber eine der Schwächen des Willke Textes. Eine Fülle an excellenten Analysen und Ideen, diese stellenweise aber versteckt in furchtbaren Anhäufungen komple-

xer Sätze und einer Explosion des Fremdwörterlexikons. Üblicherweise würde man dies von Texten erwarten, die inhaltlich nichts zu bieten haben, und durch pseudo-intellektuelles Getue „hervorstechen“ oder besser blenden wollen. Dies hat der Text aber eigentlich nicht nötig.

Gut nimmt man dies zur Kenntnis und stellt fest, daß Dystopie bei Willke *nicht* bedeutet: die ins negative verkehrte Utopie, also etwa Georg Orwells 1984, so ist dies auch schon die einzig verständliche Aussage, die Willke über Dystopien zu entlocken ist (der Verweis bzw. die Bezugnahme auf den Begriff Atopia macht nur Sinn, würde dieser auch definiert werden...). Folglich interpretiere ich *Dystopie* eigenmächtig als die grauen Seiten oder blinden Flecke einer Gesellschaft, die klassische und einfache Ordnung verloren hat und deren Verständnis und Umgang mit Wissen in radikaler Weise verändert wird.

„Die gegenwärtige Moderne ist so furchtbar fortgeschritten, dass Erfolg und Krisen gleichzeitig auftreten. Ob jemand die Erfolge feiert oder die Krisen zelebriert, hängt vom kontingenten Standpunkt und der Wahl des Kanals ab.

[...]

Eine Krisis des Wissens entsteht deshalb im Kern dadurch, das sich niemand auf das vorhandene Wissen verlassen kann, solange das komplementäre Nichtwissen nicht in gleicher Weise zur Kenntnis genommen und handhabbar gemacht ist wie das Wissen selbst. Die Krisis des Wissens bezeichnet die Unfähigkeit, mit Nichtwissen kompetent umzugehen.“ *Helmut Willke [20]*

7.2 „Erfolgsfaktor“ Wissen als Ressource?

„With the advent of the third industrial revolution, skills and knowledge have become the only source of sustainable long-term competitive advantage. [...] The knowledge that used to be tertiary after raw materials and capital in determining economic success is now primary.“ *Lester Thurow [17]*

Willkes Dystopia [20] ist zweifellos aus verschiedener Hinsicht ein sehr wesentlicher Text. Die systemtheoretischen Betrachtungen von Wissen und Nichtwissen und deren Bedeutung für Ökonomie, Wissenschaft und Gesellschaft eröffnen einige neue Sichtweisen auf die Probleme, die aus der Wissensgesellschaft und deren noch rudimentäres Verständnis durch Politik und Gesellschaft entstehen. Allerdings möchte ich die Bedeutung, die Willke *Wissen als Ressource* im Sinne des zentralen ökonomischen Erfolgsfaktors zuerkennt, kritisieren. Er schreibt z.B.:

„Dumm in einem technischen Sinne ist derjenige, der hinsichtlich der Ressource Wissen arm ist. In diesem Verständnis von Wissen sind jedenfalls in der Wissensgesellschaft die Armen die Dummen und die Dummen die Armen. Die Reichen dieser Gesellschaft sind diejenigen, die über relevante Daten, Informationen und/oder Erfahrungskontexte verfügen. Dies ist an den neuen Superreichen, den Milliardären von Microsoft, Cisco, Nintendo, Intel, Yahoo etc. besonders deutlich erkennbar, aber darüber hinaus ganz generell an den „Superstars“ der unterschiedlichsten Bereiche, von Kunst und Kultur über Pop, Sport, Medizin oder Medien bis zu Investmentbanking, Beratung oder Management. *Diese Personen verfügen über einzigartige Daten, Informationen oder Erfahrungen, also über einzigartiges*

Wissen, und das macht sie zu den Erfolgreichen und Begüterten der Wissensgesellschaft.“ Helmut Willke (Hervorhebung hinzugefügt) [20]

Dem ersten Teil ist aus meiner Sicht zuzustimmen; problematisch ist allerdings der zweite hervorgehobene Teil des Zitates. Dieser Text legt die Vermutung nahe, es gäbe so etwas wie ein „magisches Wissen“, das diesen Firmen innewohnt, und das sie unverwechselbar und stark macht. Dieses ist tatsächlich nicht vorhanden- sie kochen alle nur mit Wasser, die Großen allerdings mit wesentlich mehr Wasser als die Kleinen. Wissen hat als Ressource betrachtet an Bedeutung zweifellos die „klassische“ Ressource *Material* abgelöst (beim Kapital ist schon Skepsis angebracht) diesem Urteil möchte ich beipflichten, allerdings übersieht Willke die Tatsache, daß diese *Ressource Wissen* in einem äußerst hohem Maße flüchtig ist.

Nehmen wir als eines der genannten Beispiel Microsoft: In der Geschichte der erfolgreichen Microsoft Produkte gab es, und dies ist doch in hohem Maße erstaunlich, praktisch keine signifikanten Innovationen die aus der Firma selbst heraus entsprungen sind. Die also, im Willkeschen Sinne durch die einzigartige Erfahrung, das Wissen etc. folgen würde. Ein paar Beispiele: Graphische Betriebssysteme und Maus wurden von Xerox Parc entwickelt, und von Apple, Commodore, Atari und anderen bereits erfolgreich eingesetzt, bevor MS diese Idee übernommen und technisch eigentlich nicht konkurrenzfähig in die ersten Windows Versionen umgesetzt hat. Tabellenkalkulationen wie Excel gab es lange vorher, und wurden im Grunde von Lotus 1-2-3 kopiert, ähnliches gilt für Textverarbeitung (WordPerfect kann bis Mitte der 90er Jahre als der Konkurrenz weit überlegen betrachtet werden), für Desktop Publishing (auch hier Xerox). Datenbanken gab es vor SQLServer (Oracle, IBM), das Internet hat MS komplett verschlafen, es wurde dann Netscapes Browser kopiert, und Know-How zugekauft. Wenn Windows NT/2000 „innovativer“ als die früheren Windows Versionen sein mögen, so kaum aufgrund enormen innovativen Potentials innerhalb der Firma, sondern vorzugsweise weil Wissen in Form kompetenter Personen eingekauft wurde (bspw. von Digital, deren hervorragendes VMS Betriebssystem wiederum Pate in vielen Bereichen von NT Stand), Grafiksoftware wird zugekauft (Visio) und nur in das eigene Portfolio integriert und die „neue“ .net Strategie ist letztlich im wesentlichen eine Kopie der Java Erfolgsgeschichte „proprietarisiert“ auf Microsoft Plattform; und so könnte man diese Liste noch lange fortsetzen. Sicherlich gab und gibt es immer wieder graduelle Verbesserungen verschiedener Systeme, das ist ja keine Fragen, aber nahezu keine wesentlichen Innovationen!

Dieses Beispiel soll jetzt weniger eine Kritik an Microsoft oder deren Produkten sein, als vielmehr an der Idee der Erfolg eine Firma könnte im besonderen Wissen (eben als Ressource) stecken, die diese Firma *selbst* aufgebaut hätte. Dies trifft offensichtlich nicht zu. Tatsächlich ist das Wissen, das in vielen der *New Economy* Firmen steckt ziemlich ubiquitär verfügbar bzw. es ist ein leichtes sich das nötige Wissen anzueignen¹⁶.

Ähnliche Kritikpunkte könnte man übrigens leicht auch an den anderen zitierten Beispielen anbringen. Betrachtet man die heutige „Pop-Szene“ so zeigt sich, daß nicht Können, oder spezifisches Wissen von Relevanz sind, sondern im wesentlichen überall verfügbares oder kaufbares Wissen. Essentiell ist allerdings Marketing mit entsprechender *Marktdurchdringung* (durch Massenmedien), die aus völlig unbekanntem, die heutigen Cash-Cows der Musikszene erzeugen, ja geradezu klonen. Mögen diese nun *Backstreet Boys* oder *No Angels* heißen.

¹⁶Mit wenigen Ausnahmen. Es gibt in Bereichen, wo hoher technischer oder apparativer Aufwand o.ä. notwendig ist, um Wissen zu aquirieren oder diese Kompetenzen tatsächlich zu erwerben. Als Beispiel seinen manche Biotechnologie/Pharmazie Bereiche zu nennen, bzw. Firmen wie Intel und andere Hardwareproduzenten.

7.3 Bedingungen für ökonomischen Erfolg in der Wissensgesellschaft

Es herrscht also die paradoxe Situation vor, daß Wissen zwar einerseits die wichtigste Ressource der modernen Ökonomie darstellt und somit von äußerst hohem Wert ist. Die ökonomische Bedeutung *konkreten* Wissens jedoch aufgrund der hohen Flüchtigkeit sowie relativ leichten Möglichkeit dieses anzueignen von Willke und anderen stark überschätzt wird. Letztlich ist hier die moderne Wissensproduktion selbstverstärkend. Elektronische Publikation, einfache Zugänglichkeit von Büchern und Internet machen es im Prinzip jedermann leicht möglich sich ganz aktuelle Entwicklungen anzueignen. Die moderne kommunikations-getriebene Welt ist somit viel kleiner geworden als dies vor 50 oder 100 Jahren der Fall war. Die Bedeutung des *einzelnen* Wissenschaftlers hat folglich an Bedeutung verloren, bei gleichzeitiger Stärkung der Differenzierung (Spezialisierung) in den Fachbereichen. Dies zeigt sich auch symptomatisch an der zunehmenden Problematik Wissenschaftspreise wie den Nobelpreis zu verleihen. Gab es zu Beginn des 20. Jahrhunderts viele Einzelverleihungen, haben moderne Erkenntnisse so viele „Mütter“ und „Väter“, daß es nicht so einfach ist, konkrete Urheber bestimmter wesentlicher Entdeckungen gerecht zu definieren. So ist es auch erklärbar, daß die Nobelpreise der letzten Jahre üblicherweise an mehrere Wissenschaftler gleichzeitig verliehen wurde.

Es bleibt also die Frage offen, was nun der wesentliche Erfolgsfaktor der zitierten Firmen ist. Ich bin der Ansicht es sind *Kundenbindung, Flexibilität und Kapital*. Das Kapital der New Economy sind die Kunden, was bedeutet, daß der *Wert des Kunden* dramatisch steigt, auch wenn er sich dessen möglicherweise nicht bewußt ist. Know-How, Wissen, Produkte sind heute leicht austauschbar, mit Kapital kaufbar. Kunden aber sind dies nicht ohne weiteres. Dies beantwortet auch die Frage, wie es kommen konnte, daß AOL Time-Warner gekauft hat und nicht umgekehrt. Time-Warner hat zweifellos im traditionelleren Sinne mehr Werte, aber AOL verfügt über eine riesige Menge an Kunden, die relativ fest gebunden sind, und über die recht genaue Informationen vorliegen. Das macht sie zu einer enorm wertvollen Ressource, zumal ein Betrieb wie Time-Warner auf der anderen Seite ja seine Produkte wie Nachrichten, Unterhaltung etc. an den Mann bringen muß. Oder werfen wir einen Blick auf ein anderes Beispiele: Yahoo. Technisch gesehen ist Yahoo beliebig austauschbar. Jeder Informatiker wäre prinzipiell in der Lage in relativ kurzer Zeit eine Yahoo-artige Community zu entwickeln oder einzukaufen und zusammenzustellen, und dennoch gibt es nicht viele Communities mit dem ökonomischen Wert von Yahoo: Auch hier ist der Faktor eine gesunde Kundenbindung, die im konkreten Fall sicherlich auch darauf zurückzuführen ist, daß Yahoo die erste derartige Community im World Wide Web war, und sich gut weiterentwickelt hat.

Zum Begriff Kundenbindung zähle ich natürlich auch den Wert von Markennamen und Marketing an sich. Haben sich Kunden (auch Großkunden wie das Beispiel IBM zeigt) erst einmal an einen Namen, eine Firma gewöhnt, so ist es für Konkurrenten schwierig diese Kunden zu gewinnen: Bekanntlich ist die Aquisition eines Neukunden wesentlich teurer als das Halten eines bestehenden Kunden. Die Zukunft der New Economy Firmen liegt also nicht, wie vermutet wurde, in einem besonders einzigartigen Besitz von Wissen sondern vielmehr im Besitz eines guten oder wenigstens bekannten Namens mit einem guten Kundenbindungskonzept (dies kann auch durch die erwähnten monopolistischen Mechanismen erfolgen, siehe Microsoft und andere) und excellentem Marketing. Die Entwicklung wird also tendentiell in die Richtung laufen, daß als Kernaktivität von IBM, Yahoo, AOL, Microsoft etc. letztlich ein technischer Marketing- und Projektmanagement Betrieb mit hoher Customer-Relationship-Management (CRM) Kompetenz bleibt, und die Ressource, also das Wissen für die jeweiligen benötigten Produkt wird

einfach kurzfristig zugekauft oder auch im eigenen Haus entwickelt. Das ist eben nicht so entscheidend, weil einfach das Wissen und die Produkte an sich einigermaßen leicht austauschbar sind. Dies trifft jedenfalls so lange zu, so lange noch eine Auswahl am Markt vorhanden ist. Als Beispiel mag der Mobiltelefon Sektor gelten. In Bereichen, wo eigentlich keine Wahl mehr vorhanden ist (Microsoft, Intel, SAP), gelten etwas modifizierte Regeln insofern, als dann oft nicht einmal mehr das Produkt von besonderer Qualität sein muß.

Auch die oft zitierte Aussage in der modernen Ökonomie dominiert nicht der Große den Kleinen sondern vielmehr der Schnelle den Langsamen — oder ähnliche Aussagen — taugen als Bonmot, sind aber in der Praxis beobachtbar unzutreffend. Gemäß den obigen Ausführungen zeigt sich, daß der Große im Grunde immer im Vorteil ist. Eben aufgrund der Tatsache, daß Wissen zwar an sich ein hohes Kapital darstellt, aber in sich leicht austauschbar ist. Als gutes Beispiel dient wieder einmal Microsoft (oder auch SAP): beide haben den Internet Zug völlig verschlafen, und sogar noch weitergehend (Microsoft) dem Internet jede ökonomische Bedeutung abgesprochen. Und der schnellere war Netscape, der Browserproduzent mit einem Marktanteil von mehr als 80 Prozent bis Ende der 90er Jahre. Sehr spät hat sogar Microsoft den Wert des Internet erkannt und dann einfach jede Menge Geld — an dem als Ressource bei *den Großen* ja kein Mangel herrscht — in entsprechende Entwicklungen investiert. Das Ergebnis ist bekannt. MS Internet Explorer ist heute der dominierende Browser, mit allen bekannten Nachteilen die daraus folgen. Der Kleinere kann so schnell sein, wie er will, ist der Größere nur kapitalkräftig genug, wird er in der Lage sein den Kleinen entweder zu kaufen, oder eine parallele Entwicklung zu betreiben. Mit Hilfe der geeigneten Kundenbindung, im konkreten Beispiel: monopolistisches Betriebssystem, lassen sich dann natürlich auch sehr schnell ein Großteil der Kunden mit dem vermeintlich kostenlosen Browser beglücken.

Davon einmal ganz abgesehen, sind solche Vereinfachungen mit „schneller“ und „langsamer“ auch aus anderen Gründen problematisch: Man kann auch ganz schnell in die komplett falsche Richtung laufen, und dann sieht der Langsamere, der zunächst einmal den „Schnellläufer“ die ersten Fehler machen läßt, gar nicht mehr so alt aus. Dies ist die Lehre, die unter anderem aus dem dot.com Boom zu ziehen ist. Wären damals die „Großen“ und „Langsamen“ alle den „schnellen Gurus“ gefolgt, so wäre noch bei weitem mehr Geld vernichtet worden, als dies ohnedies geschehen ist. Auch hier erweist sich wieder: Der Größere mag langsamer sein, aber er hat aufgrund der besseren Kapitalstruktur natürlich auch den längeren Atem, kann bei kluger Beobachtung die Kleineren Konkurrenten auch mal in die Irre laufen lassen, und wenn ein kleinerer einen interessanten Weg beschritten hat, so hat der Große in den meisten Fällen sogar zu einem späten Zeitpunkt in einer schnellen Branche wie der IT alle Möglichkeiten. Auch auf die Gefahr hin zu langweilen, ist auch hier Microsoft wieder ein geeignetes Beispiel. Wie so oft wurden auch die Entwicklungen im Handheld und Mobiltelefon Bereich lange Zeit verschlafen. Der Boom des Mobiltelefonmarktes hat nun den Zenit sicherlich schon erreicht, und dennoch bringt MS erst jetzt ein eigenes Betriebssystem für Telephone auf den Markt, und die Chancen stehen nicht schlecht, daß MS auch hier wieder als letzter einige Erfolge feiern wird. (Nicht zuletzt vermutlich auch aufgrund der Tatsache, daß der US-amerikanische Markt noch nicht in dem Maße erschlossen ist, wie der europäische.)

Der zweite wesentliche Aspekt ist natürlich die bereits genannte Flexibilität. Groß sein alleine ist natürlich nicht ausreichend, wenngleich eine hervorragende Basis. Ein hoher Grad an Flexibilität ist erforderlich und ist wohl auch ein Motiv, warum die größten IT Firmen nahezu vollständig in den USA zu finden sind, wo bspw. Arbeitnehmerschutz nur eine untergeordnete Rolle spielt.

„Das sowjetische Zentralkomitee und seine Planungs-, Versorgungs- und Preisbestimmungsministerien hatten die Aufmerksamkeit auf einen Umstand gelenkt, auf den Friedrich von Hayek 1945 in einer berühmt gewordenen Schrift hingewiesen hat: auf den Umstand nämlich, dass in zentralisierten Bürokratien die *Informationswege* neigen *gefährlich lang zu werden*. Gegenüber einem „man on the spot“, also einem Wirtschaftsakteur, der unmittelbar am Marktgeschehen *spontan* und *unrestringiert* auf die Fluktuationen von Angebot und Nachfrage reagieren kann [...] würden zentralisierte Apparate notgedrungen ein *Informationsdefizit* aufweisen, das deren Konkurrenzfähigkeit entscheidend vermindert [...]“
Manfred Füllsack [4]

Genau diese Vorteile liegen im Augenblick auf der Seite der großen US Konzerne: sie sind in der Lage spontan und nahezu unrestringiert — ja sogar teilweise mit massiver Unterstützung von staatlicher Seite, aber nicht mit Regulierung(!) — zu arbeiten. Das Informationsdefizit allerdings ist nicht der Punkt, die Information ist wie gesagt, ziemlich frei verfügbar; nicht aber *Kapital* und *Flexibilität* um diese Information schnell und schlagkräftig für Erfolge zu nutzen¹⁷.

Die Macht der *Großen* geht mittlerweile in manchen Fällen sogar noch einen Schritt weiter: Oft ist es schon ausreichend, wenn ein Großer ein „neues“ Produkt ankündigt, im Wissen, daß ein *Kleiner* ein solches gerade auf den Markt bringen möchte, und schon zögern die Kunden das bereits vorhandene Produkt zu testen, sondern warten lieber gleich auf *das, was sich durchsetzen wird* — richtig erraten, auf die Lösung des Großen. Unappetitlicher Weise spielt es da nicht einmal eine Rolle, ob der Große überhaupt schon irgendeine Aktivität in die annoncierte Richtung unternommen hat; d.h. die Strategie funktioniert sogar dann, wenn die Ankündigung zeitlich mit dem *Beginn* der eigenen Entwicklung stattfindet. Somit verschaffen sich manche Monopolisten einen *Zeitvorsprung* von nahezu Jahren, und dies, obwohl (gut) funktionierende Lösungen bereits verfügbar wären!

7.4 Konsequenzen für Open Source/Protocol Projekte

In den vorigen Abschnitten wurde die Bedeutung, die Open Source und Protokoll Projekte für die offene Gesellschaft und Demokratie einnehmen schon deutlich genug hervorgehoben. Nun bleibt die Frage, was aus der Erkenntnis, daß Wissen immer mehr wie eine konventionelle Ressource behandelt wird, und als solche genannte Bedeutung gewinnt, für die OS/OP Entwicklung bedeutet. Aus dem oben Gesagten ist klar, daß das Problem nicht darin liegt, daß der OS Szene die Ressource Wissen knapp werden könnte. Das Gegenteil ist der Fall. Allerdings droht Gefahr aus einer anderen Richtung. Wenn die Bedeutung von Wissen als einer Ressource sich der Dimension von Erdöl, Stahl und Weizen annähert, bzw. diese überholt, sind ähnliche Machtkämpfe wie wir sie aus der Vergangenheit um die *alten Ressourcen* kennen, zu erwarten. Nun liegt auf der Hand, daß Wissen nicht so leicht schützbar, einsperrbar etc. ist wie eine Tonne Stahl oder Gold. Ist Wissen einmal in die „freie Wildbahn“ entkommen, ist es praktisch nicht mehr zurückzunehmen¹⁸.

Wenn nun das Wissen an sich nicht kontrollierbar ist, die Industrie aber der Ansicht ist, die hohen Investitionen in Wissen nicht beliebig für jedermann zur Verfügung zu stellen, werden

¹⁷Es ist eine interessante Frage, ob es möglich wäre, diese Flexibilität auch im europäischen Kontext zu forcieren ohne dabei die negativen Effekte sozialer Teilnahmslosigkeit mit zu importieren.

¹⁸Dies gilt nicht nur für Wissen, sondern auch für Daten oder Informationen, v.a. in digitaler Form in verstärkter Weise.

Mittel und Wege gesucht werden um wenigstens die *Anwendung* des Wissens zu Reglementieren. Nun ist dies keine neue Problematik: Urheberrecht und Patentrecht sind mittlerweile hunderte Jahre alt. Allerdings war das Wissen noch nie ein so relevanter und gleichzeitig (wie oben beschrieben) leicht flüchtiger Faktor. Dies führt dazu, daß Bestrebungen im Gange sind, Patent- und andere Schutzrechte unter dem Druck starker Industrielobbies so weit zu „stärken“, daß bei weitem über das Ziel hinausgeschossen wird. Als Stichwort seien die vieldiskutierten Softwarepatente genannt. Ohne diese hier im Detail diskutieren zu können, muß festgestellt werden, daß Patente in diesen Bereichen den ursprünglichen Sinn des Patentbesitzes, nämlich der Anreizbildung von Erfindern, bei weitem nicht mehr gerecht wird. Die tatsächliche Konsequenz ist folgende: Kleine Entwicklergruppen, Open Source Projekte etc. sind natürlich nicht in der Lage die Kosten sowie rechtlichen Umstände zur Anmeldung von eigenen Patenten zu übernehmen. Gleichzeitig werden diese Projekte immer in die Gefahr möglicher Patent-Verletzung laufen, weil der Aufwand der Überprüfung für diese Teams einfach nicht möglich ist. Patente dieser Art dienen letztlich also nur einem Zweck: die Stärke und Macht der ohnedies schon Starken und Mächtigen, sprich Monopolisten zu festigen und weiter zu stärken. Keinesfalls werden diese gesetzlichen Richtlinien den freien Wettbewerb stärken oder gar die technologische Entwicklung vorantreiben. Das genaue Gegenteil ist leider der Fall.

Ähnliches gilt für gesetzliche Richtlinien, die Urheberrechte schützen sollen, und tatsächlich weit über das Ziel hinausschießen wie der *Digital Millennium Copyright Act* in den USA und vergleichbare Gesetzesentwürfe in der Europäischen Union. Dem einzelne Künstler bringen diese Gesetze keinen Vorteil, sehr wohl aber den Medien Großkonzernen, die als einzige in der Lage sind von nun an ihre Vormachstellung nicht mehr auf innovative Produkte und Konzepte sondern auf gesetzliche Richtlinien zu begründen. Es ist wohl eine einmalige Entwicklung seit der Planwirtschaft in den ehemaligen kommunistischen Staaten, daß Inkompetenz und mangelnde Innovation nicht mehr den üblichen Marktregeln unterworfen wird, sondern vielmehr durch gesetzliche Maßnahmen einzementiert wird.

Dies ist nur aus einem Grund möglich: Das Kräftegleichgewicht ist in den westlichen Staaten einigermaßen aus dem Ruder gelaufen. Industrielobbies diktieren Gesetzgebung, Konsumenten und Bürgervertreter werden nicht angemessen einbezogen. Die neuen gesetzlichen Richtlinien dienen folglich auch nicht in erster Linie den Angestellten oder Künstler der genannten Industrien und Firmen, sondern vielmehr den Kapitalgebern, Eigentümern oder Aktienbesitzern.

Um die wichtige Offenheit in der Software und Protokollentwicklung zu garantieren sind folglich alle Schritte zu unternehmen um einen *Mißbrauch* der natürlich notwendigen Idee des Urheberrechtes Einhalt zu gebieten, und demokratiepolitische Desaster wie Softwarepatente, DMCA, TCPA oder Palladium zu verhindern, oder wenigstens den Schaden zu begrenzen.

8 Fazit

„Wenn man Kierkegaard darin folgen darf, dass die Philosophie mit dem Zweifel beginnt und ein menschenwürdiges Leben mit der Ironie, dann spiegelt sich in der Neuen Ökonomie die beste aller möglichen Welten, obwohl wir offiziell aufgegeben haben eine solche Welt zu suchen.“, Helmut Willke

Letztlich ist es eine gesellschaftspolitische Entscheidung welche Rahmenbedingungen man der Wirtschaft vorgibt. Dies ist aktiv steuerbar und nicht schicksalhaft gegeben! Auch ein vorausseilender Gehorsam europäischer Gesetzgebung nach amerikanischen Wünschen darf nicht

Teil einer zukünftigen europäischen Verfassung sein; gerade die USA führt ihre Wirtschaftspolitik weltweit mit eiserner Hand — es ist also kein Grund zu finden, es in der EU anders zu gestalten. Es stehen die Möglichkeiten zur Wahl, der Industrie die Verwendung offener Protokolle und Standards vorzuschreiben, jedenfalls im Rahmen der Ausschreibungen staatlicher oder staatsnaher Bereiche, bzw. in gewissen Bereichen der Gesetzgebung (Gesundheit, Lehre, Forschung, etc.) oder den Monopolisten freie Hand zu lassen alleine ihre eigenen Vorteile durchzusetzen; möglichst auch noch mit gesetzlicher Unterstützung zum Schaden letztlich aller anderen Wirtschaftstreibenden und der Gesellschaft an sich..

Es ist wichtig zu verstehen, daß Förderung von Open Source Software sowie offenen Protokollen keiner (Computer)firma schadet außer solchen, deren Produkte sich nicht durch besondere Qualität oder vernünftiges Preis/Leistungsverhältnis auszeichnen, sondern vorzugsweise durch monopolistische Marktdominanz und Ausnutzen von Seiteneffekten¹⁹. Die Öffnung hingegen fördert Innovation, neue Firmengründungen und gibt letztlich allen Marktteilnehmern gleiche Chancen hochwertige Produkte zu günstigen Preisen zu erzeugen und erlaubt gleichzeitig Offenheit und Transparenz in der von Informationstechnologie durchdrungenen Gesellschaft.

Eine zweite wesentliche Frage ist, folgt man der Erkenntnis, daß *Wissen* die ökonomische Ressource der *Wissensgesellschaften* des 20. und 21. Jahrhunderts sind, ob es gesellschaftspolitisch Wünschenswert ist, wenn Kernkompetenzen und Kernforschung sich nur mehr oder weitgehend in Industrielabors finden. Es darf dann niemanden wundern, wenn Firmen, die hohe Summen in die Aquisition dieses Wissens investiert haben, dieses dann auch wie eine wertvolle Ware entsprechend restriktiv behandeln. Es steht die politische Entscheidung an, zu definieren *was genau* gesellschaftliche Kernkompetenzen sind und sobald diese Definitionen geschaffen wurde sind diese wieder im Sinne einer *Open Ressource* in das Gemeinwesen z.B. in Form von universitärer Forschung zurückzuführen. Besonders gilt dies eben für die genannte IT Infrastruktur, aber genauso für Bereiche der Pharmazie oder Genetik. Es kann wohl nicht wünschenswert sein, daß private Investoren salopp formuliert den Menschen patentieren, und die Menschen weltweit letztlich für ihre eigene genetische Information indirekt zur Kasse gebeten werden.

Unter dem Strich bleibt uns also die Wahl eines kurzfristig etwas beschwerlicheren Weges zu Selbstbestimmtheit und Freiheit (die noch nie kostenlos zu haben war) oder der kurzfristigeren aber im Moment etwas bequemere Weg in die Abhängigkeit von wenigen Monopolisten — dies ist natürlich gleichzusetzen mit der Bankrott-Erklärung der Politik.

9 Anmerkungen zu Autor und Dokument

Ich beschäftige mich seit einigen Jahren im universitären sowie kommerziellen Umfeld mit der Entwicklung, Implementierung Anpassung und Schulung von Kommunikations-Infrastruktur und offenen Systemen. Alle eigenen Forschungen und Entwicklungen basieren auf Open Sour-

¹⁹Wenn der Marktführer im Bereich der Computer-Betriebssysteme bspw. Multimediatools oder Webbrowser „kostenlos“ mit dem Betriebssystem mitliefert, so werden natürlich andere auch bessere Produkte am Markt Schwierigkeiten haben. Der kurzfristige Konsument erkennt natürlich nicht, oder zu spät, daß er für die vermeintlich kostenlosen Programme in Wahrheit einen sehr hohen Preis zahlt — spätestens ab dem Zeitpunkt wo die Konkurrenten vom Markt verdrängt sind, und die mangelhafte Qualität der mitgelieferten Produkte die Sicherheit des Systems sowie seiner Daten ernsthaft gefährden oder die ökonomischen sowie politischen Kosten des Gesamtsystems zum Zeitpunkt der nahezu totalen Marktdurchdringung auf ein nahezu unerschämtes Niveau angehoben werden, weil bekannt ist, daß der Kunde ohnedies keine Wahl mehr hat.

ce und Open Protocol Systemen und erzeugen Open Source Systeme unter Verwendung standardisierter offener Protokolle, soweit dies möglich ist. Folglich ist auch dieser Artikel an sich Public Domain. Jeder, der sich dafür interessiert möge ihn auf beliebige Weise vervielfältigen und verteilen, sofern dies nicht mit kommerzieller Absicht geschieht. Auch Veränderung bzw. Verwendung von Teilen dieses Artikels ist gestattet, sofern im ersten Fall klargestellt ist, welche Teile verändert wurden und im zweiten Fall ein Zitat auf mich als Quelle nicht vergessen wird. Für den Fall der Verwendung oder Modifikation würde ich mich freuen, davon informiert zu werden, dies ist natürlich keine Bedingung.

Dieses Dokument wurde selbstverständlich auch mit Open Source Werkzeugen erstellt und verbreitet. Zu nennen sind: Linux als Betriebssystem, JEdit (<http://www.jedit.org>) und vim (<http://www.vim.org>) als Texteditoren, CVS als Versionierungs-Werkzeug sowie LaTeX und BibTeX (<http://www.dante.de>) [5] als Textsatzsystem und Literaturverwaltung. Als Webserver dienen Linux/Apache/Tomcat.

Wenn Sie Anregungen oder Kritik zu diesem Dokument haben, bzw. mit mir in Kontakt treten wollen, so finden Sie alle notwendigen Informationen auf meiner Webseite: <http://www.schatten.info>.

Literatur

- [1] *Apache Project*. <http://www.apache.org>.
- [2] *Bundestag.de Pressemitteilung: Empfehlung für künftige IT Ausstattung*. http://www.bundestag.de/aktuell/presse/2002/pz_0202285.html, feb 2002.
- [3] DAVIS, CHAD und COSKUN BAYRAK: *Open source development and the World Wide Web: a certain tension*. ACM SIGSOFT Software Engineering Notes, 27(5):93–97, 2002.
- [4] FÜLLSACK, MANFRED: *Leben ohne zu arbeiten? Zur Sozialtheorie des Grundeinkommens*. Avinus Verlag, Berlin, 2002.
- [5] KOPKA, HELMUT: *LaTeX Einführung*. Pearson Studium, Dritte Auflage, 2002.
- [6] KRECHMER, KEN: *Cathedrals, libraries and bazaars*. In: *Proceedings of the 17th symposium on Proceedings of the 2002 ACM symposium on applied computing*, Seiten 1053–1057. ACM Press, 2002.
- [7] MOCKUS, AUDRIS, ROY T FIELDING und JAMES D HERBSLEB: *Two case studies of open source software development: Apache and Mozilla*. ACM Transactions on Software Engineering and Methodology (TOSEM), 11(3):309–346, 2002.
- [8] *Mozilla Project*. <http://www.mozilla.org>.
- [9] O'REILLY, TIM: *Lessons from open-source software development*. Communications of the ACM, 42(4):32–37, 1999.
- [10] OUSTERHOUT, JOHN: *Free software needs profit*. Communications of the ACM, 42(4):44–45, 1999.

- [11] RAYMOND, ERIC S.: *The Cathedral and the Bazaar*. O'Reilly UK, Feb 2001.
- [12] SCHEIDER, BRUCE: *About Microsofts Trustworthy Computing Announcements*. <http://www.counterpane.com/crypto-gram-0202.html>.
- [13] SCHMIDT, JÜRGEN: *Nicht trustworthy, Internet Explorer gefährdet Rechner und Netze*. ct', 25:100–101, Dez 2002.
- [14] *Pressemitteilung der Stadt Schäßbisch Hall: Schwäbisch Hall setzt komplett auf Linux*. <http://www.schwaebisch-hall.de>, nov 2002.
- [15] *Sourceforge Open Source Repository*. <http://www.sourceforge.net>.
- [16] *Against TCPA*. <http://www.againsttcpa.com>.
- [17] THUROW, LESTER: *Building Wealth: The new rules for individuals, companies and nations in a knowledge-based economy*. Harper Collins, New York, 1999.
- [18] TORVALDS, LINUS: *Just for Fun: The Story of an Accidental Revolutionary*. HarperBusiness, June 2002.
- [19] *World Wide Web Consortium*. <http://www.w3c.org>.
- [20] WILLKE, HELMUT: *Dystopia, Studien zur Krisis des Wissens in der modernen Gesellschaft*. Suhrkamp Taschenbuch Wissenschaft, Erste Auflage, 2002.
- [21] YAMAUCHI, YUTAKA, MAKOTO YOKOZAWA, TAKESHI SHINOHARA und TORU ISHIDA: *Collaboration with Lean Media: how open-source software succeeds*. In: *Proceeding of the ACM 2000 Conference on Computer supported cooperative work*, Seiten 329–338. ACM Press, 2000.