

Zehn Mythen über Open Source Software

Von [Tim O'Reilly](#)

Aus dem Amerikanischen von [Reinhard Gantar](#)

[Zum Original](#)

Anmerkung des Herausgebers: Das folgende ist eine Niederschrift einer Rede, die Tim O'Reilly im Oktober vor einer Gruppe von Fortune 500 Executives gehalten hat.

Ich freue mich sehr hier zu sein -- sowohl, um etwas von Ihnen zu erfahren, als auch, um Ihnen meine Beobachtungen und Überlegungen mitzuteilen. Ich hoffe, Sie zögern nicht, mich durch Fragen und Kommentare zu unterbrechen, so daß wir rasch zu Themen vorstoßen, die uns alle interessieren. Ich habe versucht, meinen Vortrag nach einer Art "Top Ten-Liste" zu formulieren, um nicht erst am Schluß Gelegenheit zu bieten einzuhaken.

Das Hauptthema der Diskussion ist Open Source-Software. Nach dem großartigen Erfolg des Red Hat-Börsenganges ist Linux in aller Munde: ist es wie beim Internet, das auch aus dem "Nichts" kam und unser aller Leben verändert hat? Ich denke, so ist es. Ich denke, so ist es. Ich glaube nämlich, daß es eine natürliche Fortsetzung der Internet-Tschichte ist. Dieser Richtung folgend, möchte ich, nach der Erörterung von Linux und Open Source über einige Wege sprechen, auf denen das World Wide Web das gesamte Computerparadigma verändern wird.

Lassen Sie mich aber mit einigen Mythen über Open Source-Software beginnen.

Mythos Nr. 1. Es geht ausschließlich um Linux kontra Windows; Red Hat ist dabei nur noch ein weiterer Gegner Microsofts.

Mit meinen heutigen Ausführungen hoffe ich, Sie von der Unrichtigkeit der Vorstellung überzeugen zu können, daß es bei Open Source um ein Linux/Windows-Wettrennen geht. Ich werde zeigen, wie Open Source am Aufstieg des Internets beteiligt war, daß Open Source der Geschäftsgegenstand vieler der aufregendsten und innovativsten heutigen Technologiefirmen ist, und warum auch Sie die Open Source-Methode in Ihren Unternehmen anwenden können -- unabhängig davon, ob Ihre Firma Linux einsetzt oder nicht.

Eine Redensart besagt, daß jemand der nur einen Hammer hat die ganze Welt als Nagel ansieht. Eines der größten Hindernisse zum Verständnis des Phänomens Open Source ist, daß man versucht ist, es nach den Begriffen der vorigen Computing-Generation zu deuten und so den realen und bereits breit wirkenden Effekt zu übersehen. Ich hoffe, heute einige Teile dieses Effekts skizzieren zu können.

Mythos Nr. 2. Bei Open Source-Software gibt es weder Zuverlässigkeit noch Support.

Wenn Open Source-Software für die Verwendung zu unzuverlässig ist, dann ist auch das Internet zu unzuverlässig, denn es beruht in höchstem Maß auf Open Source-Software. Jede einzelne Internet-Adresse - sowohl Web- als auch e-Mail-Adresse - hängt vom Domain Name System, oder DNS, ab. Das Fundament des DNS ist ein Open Source-Programm namens BIND, das ursprünglich an der University of Berkeley entwickelt wurde, aber seitdem von seinen Entwicklern als frei zugängliches Softwareprojekt unter der Ägide einer Gruppe geführt wird, die als Internet Software Consortium bekannt ist. Wenn man sich die heutige Bedeutung des Internet vor Augen führt, ist BIND nachweislich eines der wichtigsten Programme der Welt.

Das ist aber noch nicht alles. Praktisch jede e-Mail, die über das Internet geschickt wird, verwendet sendmail, einen Mail Transport Server, der auf drei von vier aller Internet-Sites läßt (darunter auch denen großer Firmen, die nicht einmal wissen, daß sie sendmail verwenden). Da e-Mail-Sendungen immer von zumindestens zwei Mail-Servern behandelt werden, ist es sehr wahrscheinlich, daß beinahe jede e-Mail von sendmail abhängt. Wie auch BIND wurde sendmail ursprünglich an der Universität von Berkeley geschrieben und seither von der Entwicklergemeinde um den ursprünglichen Autor, Eric Allman, weitergepflegt.

Es ist auch wohlbekannt, daß der Open Source-Web-Server Apache mehr als 60 Prozent der Websites der Welt beherbergt, darunter auch viele der bestbesuchten, wie etwa Yahoo!, die auf einem Netzwerk von mehr als 2000 FreeBSD-basierten Maschinen und einer modifizierten Apache-Version läuft.

Dazu kommt, daß Yahoo! für bessere Skalierbarkeit und Performance Teile ihres Netzwerks an den Caching-Service-Provider Akamai geoutsourcet hat, der nicht zufällig einer von VA Linux Systems größten Kunden ist. (Ihr Briefing Paper zu diesem Vortrag konzentrierte sich auf das Fehlen von 32-Way SMP bei Linux; das Web bietet aber für Lastverteilung andere Ansätze, die die Services näher an den Kunden rücken. Das Argument mangelnder Skalierbarkeit wurde auch gegen den PC verwendet. Es ist zwar richtig, daß der PC die Mainframes nicht verdrängt hat, aber er bewirkte etwas Wichtigeres: Er schuf neue Arten von Anwendungen. Und ich glaube, das wird auch die Schlüsselrolle der Open Source-Software werden.

Bemerkenswert ist hier auch die Art und Weise, wie das Internet die Vorstellung von Zuverlässigkeit neu definiert hat: nicht als Fehlen von Ausfällen, sondern als Toleranz dagegen. Es gibt immer mehrere Wege, um ein Ziel zu erreichen. Es gibt die Möglichkeit, fehlgeschlagene Transaktionen wieder und wieder zu versuchen, und so weiter.

Das ist aber noch nicht alles! (Ich fühle mich wie ein Verkäufer von Gurkenhobeln). Der TCP/IP-Protokollstack, der von den meisten kommerziellen Internet-Softwarepaketen verwendet wird, basiert auf Code, der ursprünglich für das Berkeley Networking Package geschrieben wurde. Auch die Internet Engineering Task Force oder IETF - jene Körperschaft, die alle Internet-Standards beschließt - operiert nach einem Prozeß, der mit Open Source viel gemeinsam hat. Jeder kann der Mailing-Liste beitreten, an Meetings teilnehmen und die Standards mitgestalten -- technische Beiträge, nicht politische Manöver großer Firmen sind es, die bestimmen, was aufgenommen wird und was nicht.

Es ist auch nicht so, daß es sich bei dieser Internet-Software um kurzlebige Eintagsfliegen handelt. Viele dieser Programme wurden und werden beständig weiterentwickelt, verfeinert und über Jahre hinweg betreut -- Jahre während derer viele angeblich voll unterstützte kommerzielle Pakete im Mistkübel gelandet sind, nachdem ihre Hersteller gekauft wurden oder einfach vom Markt verschwanden.

Einer der spannendsten Aspekte der Open Source ist aber, daß sie für eine gewaltige Machtverschiebung von den Herstellern hin zu den Benutzern steht, die nicht ohne Hilfe bleiben, wenn der Entwickler das Handtuch wirft. Der Webserver Apache ist das beste Beispiel für diesen Vorzug: Nachdem die entwickelnde Universität den NCSA Webserver aufgab, weil sein gesamter technischer Stab von Netscape weggeheuert wurde, war es eine Gruppe von Benutzern des NCSA-Servers, die eine informelle Gruppe ins Leben rief (und durch eine Internet-Mailing-Liste zusammenhielt), um ihre Updates und Patches zum NCSA-Code zu koordinieren. Das war auch der Ursprung des Namens -- es war "a patchy server".

Durch wachsendes Interesse an Open Source-Software in der Computerindustrie werden große institutionelle Endbenutzer bald das beste beider Welten haben, die informellen Support-Mechanismen der vernetzten Entwicklergemeinde und den Big Company-Support von Kalibern wie IBM, HP, Sun und Silicon Graphics, von aggressiven Startups wie Red Hat gar nicht zu reden.

Mythos Nr. 3. Große Firmen verwenden keine Open Source-Software.

Trotz anderslautender offizieller Verlautbarungen gibt es eine enorme Menge an Open Source-Arbeit in Ihren eigenen Firmen. Um Ihnen nur ein Beispiel zu geben: Die von meiner Firma verlegten Bücher über Perl, die führende Open-Source-Programmiersprache, wurden in großen Mengen an praktisch jede Wall Street-Investmentbank verkauft. Eine große Zahl der Teilnehmer der Perl-Konferenzen meines Verlages kommen von Boeing und anderen Luftfahrtsunternehmen. Als Amazon kürzlich ihr Purchase Circle-Feature einführte, das die Bücher listet, die von verschiedenen Unternehmen gekauft werden, zeigte sich, daß Bücher über Perl die Top Ten jeder großen Halbleiterfirma dominiert und in mehr Purchase Circles auftauchten als jedes andere Technologiethema. Ich wette, daß wir Bücher über Perl an praktisch jede Firma in den Fortune 500s verkauft haben. Die Benutzer verwenden Perl, weil es gestattet, eine neue Klasse von Problemen zu lösen.

Anekdoten wie diese gibt es in der Open Source-Gemeinde viele. Greg Olson, der CEO von Sendmail, Inc., erzählte einmal von einem Gespräch mit dem IS Department einer großen Bank. Er hatte bemerkt, daß der Charakterstik nach der Mail dieser Bank dort sendmail verwendet wurde und fragte, ob er sie als Referenz verwenden könne. Sie leugneten aber, daß sie sendmail verwendeten, bis er ihnen die Spuren zeigte, die ihre e-Mail im Cyberspace hinterließ

Das bringt uns zu einem damit zusammenhängenden wichtigen Punkt: Die Vernetzung durch das Internet verändert gerade die Spielregeln, wieviel Information Marktforschern und den Mitbewerbern zugänglich ist. Gideon Gartner, der

Gründer des Marktforschungsunternehmens The Gartner Group, verriet mir einmal das Geheimnis seines Geschäfts: Er konnte sich zwischen die Computerhersteller, die Wallstreet-Investmentbanker und die großen Abnehmer schalten und verkaufte jedem von ihnen, was er von den anderen erfahren konnte. Das Internet hat dieses Verhältnis verändert, denn nun sind viele Technologieentscheidungen für jeden zu sehen. Während beispielsweise eine Gartnerstudie in Gesprächen mit IT-Managern erheben kann, daß der Trend bei Webservern zu Ungunsten von Microsoft in Richtung Netscape geht, kann gleichzeitig die unparteiische, automatisierte Netcraft-Zählung ergeben, daß Apache am Markt dominiert. Viel wird über die Spuren gesprochen, die wir im Cyberspace hinterlassen und die den Schutz unserer persönlichen Daten gefährden, ich glaube aber, daß wir auch enorme Auswirkungen auf dem Gebiet der IT-Marktforschung erleben werden.

Mythos Nr 4. Open Source ist der Feind des geistigen Eigentums.

Ich bin sicher, daß Sie die Schauergeschichten über die General Public License (oder GPL) gehört haben, wenn Sie auch nur ein wenig in dieses Thema eingedrungen sind. Sie hörten, daß Software unter der GPL von Ihrer Firma fordert, alle ihre Software zu verschenken.

Es stimmt zwar, daß Software, die GPL-basierten Code beinhaltet, unter denselben Auflagen veröffentlicht werden muß -- das ist die sogenannte "virale" Charakteristik der GPL. Man aber sogar so lizenzierte Software frei verwenden. Darüber hinaus ist die Free Software Foundation nur eine der Traditionen, auf denen die Open Source-Bewegung basiert. Viele der ganz wichtigen Programme, die ich vorher erwähnt habe, stammen aus der universitären Tradition der Software-Entwicklung, deren Hauptast das berkeleyische Unix-Projekt ist. Diese Programme werden unter Lizenzen veröffentlicht, die proprietäre Erweiterungen gestatten. Viele der Key Developers haben nach einer Balance gesucht, zwischen dem, was sie hergeben und was sie proprietär halten wollen.

Dazu kommt, daß viele Firmen wie etwa Netscape, IBM, Apple und viele kleinere Entwickler Lizenzen geschaffen haben, die dem Schöpfer der Software eine Reihe von Urheberrechten vorbehalten, den Quellcode aber offengelegt haben, um außenstehende Entwickler einzubinden.

Wir sind noch immer dabei herauszufinden, welche Arten von Lizenzen am besten geeignet sind, um zur Open Source-Entwicklung zu ermuntern.

Mythos Nr. 5. Open Source dreht sich nur um Lizenzen.

Richtig ist, daß es bei Open Source um Internet-basierte Kooperation geht. Lizenzen spielen nur so weit eine Rolle, als daß sie die Regeln bestimmen, die verhindern sollen, daß Firmen die Idee dahinter unterminieren.

Im Kern ist Open Source eine Methode zur Software-Entwicklung, keine politische Ideologie. Einige Prinzipien, die dieser Methode zugrunde liegen, werden in Eric S. Raymonds neuem Buch beschrieben, von dem ich ein Exemplar für jeden von Ihnen mitgebracht habe. Obwohl Eric viele dieser Prinzipien ganz richtig erkannt hat, glaube ich, daß wir die Regeln dieses neuen Spiels laufend neulernen müssen.

Versuche wie Suns Community Source License sollen die Grenzen dieses Phänomens ausloten. Welche "Regeln" sind erforderlich, um in einer Open Source-Gemeinde Interesse zu wecken, und welche Überzeugungen können dem zuwider laufen?

Die wirklichen Lektionen, die man von Open Source-Gemeinden lernen kann, sind aber die Techniken der vernetzten Zusammenarbeit, die sie als Pioniere erforscht haben. Open Source-Softwareprojekte haben Techniken entwickelt -- Verwendung von Mailing-Listen, verteilter Zugriff auf Versionskontrollsoftware, Techniken zur Kritik durch Gleichgesinnte ("Peer Review"), zur Erörterung und Abstimmung von Features, rasches Reagieren auf User Feedback und Gelegenheiten zur Mitbestimmung durch Benutzer -- die alle fruchtbar auf die Entwicklung jeglicher Art von Software angewendet werden können.

In diesem Zusammenhang möchte ich eine Firma erwähnen, die wir kürzlich zusammen mit Brian Behlendorf, dem Mitbegründer der Apache Group, ins Leben gerufen haben. Sie heißt collab.net und versucht gezielt die verschiedenen Arten von Infrastruktur-Services für Projekte zu schaffen, die auf vernetzter Zusammenarbeit beruhen. Ihr erstes Projekt ist sourceXchange, eine Art Börse für Open Source-Entwicklungen, bei der Firmen RFPs (Requests for Projects) für Software deponieren können, die sie geschrieben haben wollen. Unabhängige Entwickler können dann diese Projekte ersteigern. Auf lange Sicht will collab.net outgesourcete Services bieten, die es jeder Art von Softwareprojekt gestattet, die Techniken zu verwenden, die vom Apache-Projekt so erfolgreich eingesetzt werden.

Mythos Nr. 6. Wenn ich meine Software der Open Source-Gemeinde aushändige, werden plötzlich Tausende von Entwicklern gratis für mich arbeiten.

Wie Netscape durch das Mozilla-Projekt herausfand, gibt es keine magische Gemeinde, die nur darauf wartet, in jedes beliebige Open Source-Projekt hineinzuspringen. Linux kann Tausende von aktiven Entwicklern für sich reklamieren, weil es ein Zusammenschluß von Hunderten von unabhängigen Projekten ist.

Die meisten Open Source-Projekte haben einen Kern aus wenigen Dutzend interessierten Teilnehmern, die Berichte über Probleme, Bug Fixes und gelegentliche Ergänzungen liefern, und Tausende oder Zehntausende Benutzer. Beachten Sie aber, daß einige Benutzer schließlich von den äußeren Zonen in die inneren wandern können. Es ist der Geschichte nicht unähnlich, die ich einmal von einem Spendensammler für die Nature Conservancy gehört habe, die so um die 750.000 Mitglieder hat. Dort macht man das Geld nicht durch die Masse einzelner Mitglieder -- diese Mitglieder stellen nur den Pool, aus dem schließlich die großen Spender kommen.

Die wichtigere Lektion hier ist aber, daß, wenn man die Zusammenarbeit einer Gemeinde in Anspruch nehmen will, man die Gemeinde seiner Benutzer in Anspruch nehmen muß, keine vage definierte "Open Source-Gemeinde".

Mythos Nr. 7. Open Source hat nur für Programmierer Bedeutung, da die meisten Benutzer ohnehin niemals den Quellcode ansehen.

Die Vorzüge der Open Source sind genau die selben wie die Vorzüge jedes anderen freien Marktes: Wettbewerb zwischen mehreren Anbietern bewirken niedrigere Preise, mehr Innovation und Spezialisierungen, die den Bedarf neuer Nischen bedienen.

Das ist meiner Meinung nach der wirkliche Hauptvorteil der Open Source: Man ist nicht mehr länger einem einzigen Anbieter ausgeliefert. Wenn das Schlimmste passiert, kann man seine Probleme selbst lösen, oder, was wahrscheinlicher ist, sich an mehrere Anbieter wenden, die einen besser betreuen können, als das ein einzelner Anbieter jemals können wird.

Kurz gesagt, Open Source hat nicht nur für Software-Hersteller Bedeutung. Es ist eine Möglichkeit für Sie, Ihren internen Benutzern und Kunden bessere Dienstleistungen zu bieten, und das durch vernetzte Kooperation, wie sie von den besten Software-Entwicklern erprobt wurde.

Mythos Nr. 8. Mit freier Software kann man kein Geld verdienen.

Die Aufregung um Red Hat und ihr Erfolg beim bloßen Verschieben von Software in Schachteln verstärkt die Illusion, daß die meiste Software für den Weiterverkauf als Ware geschrieben wird. Tatsächlich ist es aber so, und das wissen die meisten von Ihnen schon, daß man Software für den Gebrauch im eigenen Unternehmen schreibt. Sie ist ein Werkzeug wie jedes andere und wie überall gibt es Nachteile und Vorteile beim Kauf und beim Selbermachen.

Es stimmt, daß Open Source-Software die Summen verringern wird, die für existierende kommerzielle Software ausgegeben werden (was für Hardware-Hersteller ein Grund ist, auf den Open Source-Zug aufzuspringen, immerhin umgeht man die "Microsoft-Steuer"). Bahnbrechende Technologien wie Open Source-Softwareentwicklung reduzieren die Gewinnspannen der etablierten Mitspieler, senken die Hürden für Innovation und erweitern den Markt für Anbieter, die eine rasche Auffassungsgabe haben und bald nach den neuen Regeln spielen können.

Eine historische Parallele ist die Geschichte der PC-Industrie. Im Prinzip änderte IBM durch die Freigabe der Spezifikation des Personal Computer als offener Standard die Spielregeln. Jahrelang gab es eine sehr leicht durchschaubare Schlacht um proprietäre Standards, aber schließlich wurde klar, daß auf Systemebene der strategische Vorteil nicht im Erringen von proprietären Privilegien lag, sondern im besseren Management der Herstellungs- und Vertriebslogistik. Deshalb ist Dell, eine Firma, die von einem College-Studenten gegründet wurde, heute ein Multimilliarden-Dollar-Unternehmen. Deshalb behauptet Bob Young von Red Hat, daß seine Firma keine reine Software-Firma ist, sondern eine Brand Marketing Company, und er besteht darauf, daß es sein Ziel sei, "den Markt für Betriebssysteme zu verkleinern". Sobald sich bestimmte Arten von Software verändern, verändern sich auch die Fähigkeiten entsprechend, die notwendig sind, um Erfolg zu haben.

Das Internet, eine bahnbrechende Technologie, die auf offenen Standards und Open Source-Software beruht, hat riesige neue Märkte außerhalb der Software-Industrie erschlossen. Ich habe Yahoo! erwähnt. Man kann nachweisen, daß die Explosion des Web eine unmittelbare Folge der Open Source-Bewegung ist. Ich meine damit nicht nur die Rolle von freien Betriebssystemen wie FreeBSD und von freien Webservern wie Apache. Der Umstand, daß Apache drei Millionen Sites für sich verbuchen kann und daß die Suchmaschinen des Web überhaupt indizieren können, beruht auf der Tatsache, daß HTML selbst ein offener Standard ist. Viele Leute, die Web-Pages gestalten, tun das durch Nachahmung -- die Web Browser verfügen über einen Menüpunkt "Quellcode anzeigen", der zeigt, wie jedes Feature implementiert ist, und das erleichtert es, sie zu kopieren. Das ist Open Source in Aktion, Arbeit an der Basis. So kann ich hier ein kraftvolles Argument anführen, das die Gründer von Yahoo! wohl unterschreiben würden: Yahoo! und viele andere prominente Internet-Erfolge sind eigentlich Open Source-Erfolge. Als einer der Leute, die die Web-Revolution in Gang gesetzt haben, kann ich persönlich die Wichtigkeit von Open Source in unseren frühen Unternehmungen nur bestätigen.

Ein Beispiel dafür, wie man außerhalb der üblichen Kategorien darüber nachdenkt, wie man durch Open Source-Software Geld verdienen kann, liefert die Internet Service Provider-Industrie. Wenige Leute realisieren, daß Rick Adams, Gründer des ersten kommerziellen ISPs UUNET, ein anerkannter Autor freier Software war. Er schrieb sowohl B News - zu jener Zeit der dominierende Usenet News Server - als auch die sehr verbreitete SLIP-Implementation, die Internet-Zugang über Dial Up-Modems gestattete. Rick versuchte aber erst gar nicht, seine Software zu verkaufen: Er erkannte, daß das Geld im Aufbau von Diensten lag, die für die Verbreitung dieser Technologie im Mainstream jenseits der Hacker-Elite notwendig war.

Mythos Nr. 9. Die Open Source-Bewegung ist nicht von Dauer; die Leute werden aufhören, freie Software zu entwickeln, sobald sie sehen, daß andere viel Geld mit ihrer Arbeit verdienen.

Wenn man sich die Entwicklergemeinden um die meisten Open Source-Projekte ansieht, dann sieht man ein sehr großes Kontingent von Leuten, die Open Source-Projekte finanzieren, weil sie die Software bei ihrer Arbeit verwenden wollen oder einen anderen Weg gefunden haben, damit Gewinne zu machen.

Ich habe vorher erwähnt, daß das Apache-Projekt von einer Gruppe von Anwendern des NCSA-Servers gegründet wurde. Das ist nicht ganz richtig, denn einige dieser "Anwender" waren Web-Design- und Hosting-Firmen, die ihre Services weiterverkauften. Zugriff auf den Server-Code zu haben, war für ihr Geschäft lebenswichtig und deshalb war es sinnvoll, die Weiterentwicklung zu bezahlen. Durch Kooperation bei den Verbesserungen waren sie in der Lage, sofort Wettbewerbsvorteile zu erzielen und ihren Kunden neue Features zu bieten. Und da ihre Dienstleistungen in der Regel ortsgebunden waren, machte es sogar Sinn, mit Firmen im selben Business zusammenzuarbeiten.

Die Key Developers der meisten heutigen Open Source-Projekte sind eine Mischung aus Universitätsforschern, Entwicklern in Firmen, die ein bestimmtes Open Source-Paket bei ihrer Arbeit verwenden, unabhängigen Consultants, die von der erhöhten Aufmerksamkeit profitieren, die ihre Teilnahme bewirkt, und Entwicklern, die von Firmen gesponsert werden, die mit diesem Projekt Umsatzerwartungen verbinden.

Mythos Nr. 10. Open Source kann nur imitieren, was Microsoft und die kommerzielle Welt erfinden.

Zwar ist es richtig, daß es bedeutende Anstrengungen in der Linux-Welt gibt, Äquivalente des Windows-Desktops und der üblichen Office-Applikationen zu schaffen, aber das sind nicht die wichtigsten Schauplätze des Open Source-Phänomens.

Denke Sie für einen Moment an die aufregendsten neuen Computeranwendungen für Konsumenten. Das sind nicht mehr länger Desktop-Anwendungen (man kann ohne weiteres behaupten, die letzte bedeutende derartige Anwendung wäre der Internet-Browser gewesen, seine Einführung ist aber schon sechs Jahre her). Die spannendsten Applikationen sind heute alle Web-basiert -- [Amazon](#), [eBay](#), [E*Trade](#), [maps.yahoo.com](#). Neue Funktionalität wird über das Web geliefert. Ich würde behaupten, daß sogar im Back Office das Web gerade alles verändert.

Wenn ich noch einmal die Geschichte der frühen Tage des IBM PCs betrachte, meine ich, daß es IBMs größte Leistung war, die Eintrittshürden in den Computermarkt zu senken. Sobald die Spezifikation für den PC öffentlich war, konnte jeder einen PC bauen. Und als Folge davon passierten zwei Dinge: Erstens entstand ein "Commodity Hardware Business"; viele Anbieter traten in Wettbewerb um den Computer mit dem niedrigsten Preis. Der Einstieg wurde so

leicht, daß Michael Dell sein heutiges Multimilliarden-Dollar-Business in seinem Zimmer im Studentheim starten konnte.

Vielleicht noch wichtiger war aber der Auftrieb, den die offene Hardware-Plattform der Software-Industrie gab. Plötzlich waren auch hier die Eintrittshürden sehr niedrig. Statt wie bisher bloße Satelliten der Hardware-Hersteller zu sein, wurden die Softwarefirmen selbst zu einer eigenständigen Kraft. IBMs großer Irrtum beruhte auf der falschen Annahme, daß Hardware mehr zähle als Software, was Microsoft die Gelegenheit gab, die Pole Position in der Computerindustrie zu übernehmen.

Ich behaupte, daß "Commodity Software" einer neuen Klasse von Applikationen zum Aufstieg verhelfen wird, die ich "Infoware" nenne.

Wenn man diese "Applikationen" betrachtet, sieht man, daß sie eher Prozesse als Produkte sind. Microsoft kommt mit neuen Versionen alle 12 bis 18 Monate heraus. Yahoo!, Amazon und E*Trade bringen ihre Produkte unentwegt auf einen neuen Stand. Hinter den Kulissen sieht man Open Source-Scripting-Sprachen wie Perl, Tcl und Python (oder kommerzielle Produkte, die ihre Funktionalität imitieren) als wichtigsten Teil des Entwicklungs-Mixes. Der Grund dafür war, und das haben sowohl Microsoft mit Active X als auch Sun mit klientenseitigen Java verpaßt, war, daß diese "Applikationen" nicht von Programmierern gemacht werden, sondern von Autoren, Redakteuren, Katalogisierern und anderen Content-Spezialisten. Manche dieser Programme erzeugen ein "ad hoc"-Mapping von dynamischen, textbasierten Inhalten wie etwa Nachrichtentickern. Perls Fähigkeit, Text durch mächtige Regular Expressions zu parsen, stellte sich als wichtiger für die Applikationen der Zukunft heraus als objektorientiertes Recycling von Code.

Für mich ist das die wirkliche Bedeutung der Open Source-Bewegung. Wenn man die Eintrittshürden senkt, erhöht man die Wahrscheinlichkeit von Überraschungen. Wie Alan Kay einmal sagte: "Es ist einfacher, die Zukunft zu gestalten, als sie vorherzusehen." Open Source gibt uns ein besseres Werkzeug für Innovationen in die Hand; nicht weil es eine magische Entwicklungsmethode ist (obwohl die Vorzüge verteilter Kritik durch Kollegen gewaltig sind), sondern weil mehrere Mitspieler unerwartete Wendungen hineinbringen können. Software-Firmen konnten das Web nicht erfinden, weil sie zuviel zu verlieren hatten und versuchten, ihre engstirnigen Produktauffassungen die mit ihren existierenden Ertragsmodellen kompatibel waren, in die Welt des vernetzten Multimedia zu pressen. Es war die Verfügbarkeit von freier Software und offenen Standards, die Leute außerhalb dieser Industrie in die Lage versetzte, das nächste große Paradigma zu schaffen.

Das wirkliche Geheimnis der Open Source ist, daß sie der neueste Technologie-Durchbruch ist, eine der die existierenden Anbieter entmachtet und neue Ideen hereinläßt. Das letzte Mal waren die "Barbaren" (um Phillippe Kahns Ausdruck zu verwenden) kleine Software-Firmen. Heute hat Microsoft den Software-Markt erobert und würgt alle Innovation ab, aber schon kommt die "Commoditization" der Software durch die Impulse der Open Source-Bewegung herein und ermöglicht eine völlig neue Klasse von Anwendungen.

Heißt das, die Software-Industrie, wie wir sie kennen, ist irrelevant geworden? Ganz und gar nicht. Sie wird weiterhin blühen und gedeihen, so wie es ja auch den Hardware-Herstellern in einer von Software-Firmen dominierten Epoche gut gegangen ist. In einer Hinsicht ist es irrelevant, ob das Web bei seinen offenen und freien Wurzeln bleibt; seine Mission hat es bereits erfüllt. Ich erwarte sogar, daß viele Applikationen, die ursprünglich in der Open Source-Gemeinde entwickelt wurden, irgendwann in den nächsten paar Jahren proprietär werden, weil sich viele Hersteller von Web-Applikationen, die ihren Wohlstand auf einem offenen Fundament aufbauen, sich selbst schützen werden wollen. Sogar Microsoft war einmal ein Outsider, eine kleine Firma, die die Welt verändern wollte.

Ich glaube, daß es zu einer erfolgreichen Industrie gehört, die richtige Balance zwischen offen und proprietär zu bieten. Im Kern der offenen PC-Hardware-Plattform steckt eine proprietäre CPU und eine Reihe von proprietären Geräten. Im Kern des offenen Internet stecken proprietäre Cisco-Router, und zu jedem Open Source-Programm gibt es ein proprietäres Gegenstück. Es ist keine Entscheidung entweder-oder.

Nach allem dem Gesagten glaube ich, daß wir aus unseren Fehlern lernen können. Es ist nicht nötig, daß wir periodisch durch Zyklen von Offenheit und gesundem Wettbewerb gefolgt von Stagnation durch einige wenige dominierende Hersteller gehen, die uns zentral nach ihrem Master Plan managen. Die Geschichte lehrt uns, daß, wenn es um Innovation geht, offen proprietär immer schlägt. Man muß sich nur Unix ansehen, um diesen Effekt zu beobachten. Viele der Innovationen, die in kommerzielle Unix-Systeme Eingang fanden (wie auch die grundlegenden Technologien des Internets) wurde ursprünglich von den Universitäten als Erweiterungen zur Arbeit bei Bell Labs entwickelt. Als AT&T Unix unter einer restriktiven Lizenz verkommerzialisierte, hörten diese Entwicklungen auf und fingen erst wieder an zu blühen, als Linux, eine freie Implementation, zum Unix-Marktführer wurde.

Mein Schlußwort an Sie ist daher, daß wenn Sie Wettbewerb und Innovation schätzen, es in Ihrem Interesse ist, die

Gemeinde der Open Source-Entwickler zu fördern und zu unterstützen. Sie sollten nicht nur mit Open Source-Produkte experimentieren, sondern auch von den zugrundeliegenden Prozessen lernen. Mein Traum ist, daß wir das Beste aus beiden Welten haben können: Eine lebendige kommerzielle Industrie, die dort, wo das sinnvoll ist, auf Offenheit und Kooperation basiert, und Wettbewerb und proprietäre Privilegien, wo sie Sinn haben.

Einige abschließende Gedanken

Vor circa zehn Jahren prägte Sun Microsystems den Slogan, der erst jetzt Wirklichkeit wird: "Das Netzwerk ist der Computer". Ich möchte einige Beispiele skizzieren, in denen ich das verwirklicht sehe:

- Jedes Gerät wird zur Netzwerkperipherie
- Applikationen leben im Web statt auf dem Desktop. Wir erkennen das an Suns und Microsofts Plänen, Applikationen als gehostete Services anzubieten, aber noch mehr gilt das für die neue Klasse von Anwendungen, über die ich vorhin schon gesprochen habe.
- Websites laufen nicht mehr länger auf einzelnen Servern, die von einer einzigen Firma gesteuert werden. Ich habe die Rolle von Caching Services wie Akamai bereits erwähnt. Wenn man sich die [Beta-Kapitel von O'Reilly-Büchern auf Amazon](#) ansieht, besucht man eigentlich eine O'Reilly-Site, die mit der Amazon-Site verschmolzen ist. Bei vielen Suchmaschinen verwendet man eigentlich einen remoten Suchdienst. Mapping-Services wie maps.yahoo.com sind von anderen Firmen geoutsourcet. Beim Nachschlagen einer Telefonnummer bei PageNets Zweiweg-Pager, spricht man eigentlich mit einem Proxy-Server im Web, der alles, außer der Telefonnummer verwirft und zum Pager schickt. Bei Palm VII und WAP-fähigen Telefonen sind die "Klienten" der Websites oft keine Browser, die von Menschen direkt betrieben werden, sondern die Klienten anderer Programme sind. Die Liste nimmt kein Ende.
- Eine weitere spannende Idee ist es, sich jegliche Website als mit einem impliziten API ausgestattet vorzustellen und wie eine Unteroutine eines Programmes zu verwenden. Wir machen das bereits in kleinem Maßstab im ganzen Web. Beispielsweise schrieb Jon Udell, einer unserer Autoren, ein Script, das er das "Mindshare-Script" nennt. Es beginnt mit einem beliebigen Punkt in der Yahoo!-Verzeichnishierarchie, "entfaltet ihn", um eine vollständige Liste aller Sites zu bekommen, die Yahoo in dieser Kategorie hat, und füttert diese eine nach der anderen als Suchbegriffe in Altavista. Das Ergebnis ist eine sortierte Liste aller Sites zu einem gegebenen Begriff, die in absteigender Reihenfolge angeben, wieviele und welche Sites ein Link dorthin haben. Mit der Verfügbarkeit von XML werden solche Dinge viel einfacher und robuster werden. Eines der aufregendsten Gebiete, die man im Augenblick beobachten kann, ist das des XML-RPC. XML-RPC ist in ein Protokoll namens SOAP (Standard Object Access Protocol) integriert, das sowohl von Microsoft als auch der Open Source-Gemeinde unterstützt wird. Das ist die erste Andeutung einer wahrlich globalen Veränderung in der Art und Weise, wie Software geschrieben wird.

Open Source ist wie ein Stein, der in einen Teich geworfen wird. Die Wellen breiten sich weiter aus, auch wenn man den Stein nicht mehr sehen kann, der sie hervorgerufen hat.

Ein letzter Punkt über Open Source selbst, der bei zeitgenössischen Firmen einige Gedanken über deren subversive Kraft provozieren sollte. Open Source-Projekte werden von individuellen Entwicklern gemanaget, nicht von Firmen. Wie Larry Augustin von VA Linux Systems (der mehr Open Source-Stars beschäftigt als jede andere Firma) bemerkt hat, gibt es hier eine enorme Machtverschiebung. Wenn ein führender Open Source-Entwickler das Unternehmen wechselt, geht sein Projekt (und dessen Status) mit ihm. Es drängt sich ein Vergleich mit der Filmindustrie der Dreißiger und Vierziger Jahre auf, als das Studio-Modell durch das heutige von Stars dominierte System abgelöst wurde.

Wenn diese Machtverschiebung bedeutet, daß Schlüsselprojekte von einzelnen Individuen kontrolliert werden statt durch verschiedene Personen, wäre dies ein Grund, sich vor dieser Veränderung zu fürchten. Aber die einzigartige Charakteristik der Open Source-Bewegung, die Schlüsselentwickler zwar führen, aber das Projekt nicht einseitig steuern läßt, bietet eine Versicherung sowohl gegen den Verlust dieses Entwicklers als auch dagegen, daß er das Projekt auf einen Irrweg leitet.

Kurz gesagt, die Open Source-Bewegung wird uns erhalten bleiben. Sie hatte bereits großen Einfluß, der sich aber noch erheblich vergrößern wird. Halten Sie die Augen offen und bereiten Sie sich auf noch mehr erfreuliche Überraschungen vor!