

## Geschenkökonomie im luftleeren Raum

### Anmerkungen zur Open-Source-Bewegung

Wir befinden uns im Jahre 2002. Alle Computer werden von einem Betriebssystem beherrscht. Alle Computer? Nein! Der Stamm der *Hacker*<sup>a</sup> hört nicht auf, dem unfähigen Monopolisten Widerstand zu leisten. Der Zauberspruch dieses Stammes ist der offene Quellcode der Software (Open Source).

So könnte popkulturell gedeutet eine (Selbst-)Beschreibung der Open-Source-Bewegung aussehen. Mediale Aufmerksamkeit hat diese Bewegung vor allem mit dem bekanntesten und erfolgreichsten Projekt aus diesem Umfeld, dem Betriebssystem Linux erregt.

### Wie funktioniert Open-Source-Softwareentwicklung?

Das Leitprinzip der Open-Source-Entwicklung ist die gemeinsame kooperative Entwicklung von Programmen durch das Veröffentlichen des Quellcodes, also die in einer Programmiersprache verfaßten, algorithmisierten ›Texte‹. Dieser Quellcode muß, um im Computer verwendet werden zu können, durch sogenannte Compiler oder Interpreter in Maschinensprache (Binärcode) übersetzt werden. Das Veröffentlichen des Quellcodes und die damit verbundene Erlaubnis zur freien Benutzung und Veränderung führen zu einer gegenseitigen Prüfung des Codes und zur parallelen Fehlersuche vieler Anwender. Der in kurzen Abständen veröffentlichte Quellcode wird von Entwicklern und Benutzern getestet, und vorhandene Fehler (*bugs*) oder fehlende Funktionen werden schnell gefunden oder ergänzt. Die Open-Source-Bewegung hat den Beweis geliefert, daß diese Arbeitsweise zu einer innovativen und schnellen Entwicklung von stabiler Software führt. Auch hochkomplexe Programme wie Betriebssysteme (Linux<sup>b</sup>), graphische Benutzeroberflächen wie (KDE<sup>c</sup>), der meistbenutzte Webserver (Apache<sup>d</sup>) oder Bildbearbeitungsprogramme (GIMP<sup>e</sup>) werden unter diesen Vorzeichen in einer Qualität entwickelt, die kommerziellen Produkten in nichts nachsteht, ja diese oft sogar übertrifft. Die Zugänglichkeit des Quellcodes und die weitere offene Entwicklung werden durch Lizenzen abgesichert.

### Open-Source-Lizenzen

Diese Lizenzen regeln den rechtlichen Status der *freien Software*. (Frei ist hier im Sinne der *freien Rede* gemeint, nicht im Sinne von *Freibier*, so die klassische Erläuterung). ›Open Source‹ meint aber nicht nur Zugang zum Quellcode. In den Lizenzen sind immer auch zusätzliche Kriterien festgeschrieben.

Eine Zusammenfassung der Grundsätze existierender Lizenzen und eine Basis für die Entwick-

<sup>a</sup><http://catb.org/~esr/jargon/>

<sup>b</sup><http://www.linux.de/linux>

<sup>c</sup><http://www.kde.org>

<sup>d</sup><http://www.apache.org>

<sup>e</sup><http://www.gimp.org>

lung weiterer Lizenzen will die Open Source Definition<sup>a</sup> liefern: Eine Lizenz soll demzufolge niemand am Verkauf oder am Verteilen der Software hindern. Das Programm muß dabei auch im Quellcode (mit-)verteilt werden. Änderungen oder abgeleitete Programme müssen jederzeit möglich sein. Dabei müssen vorgenommene Änderungen ausgewiesen und den Entwicklern zugeordnet werden können.

Damit werden einige herkömmliche Eigentums- und Copyrightrechte ausgehebelt. Jeder kann, sofern er der verwendeten Programmiersprache mächtig ist, die Software ändern, kopieren, verteilen oder auch verkaufen.<sup>1</sup>

## Die Eigentumspraxis

Soweit zumindest die rechtliche Theorie. In der Praxis stellt sich die Eigentumsfrage an der Software etwas anders dar. Eric S. Raymond, Hacker und Anthropologe der OS-Bewegung beschreibt in seinem Aufsatz »Homesteading the Noosphere«<sup>b</sup> diese Praxis. *Homesteading* ist ein juristischer Begriff aus der Landnahmezeit der USA im 19. Jahrhundert. Demnach hatte jeder Weiße das Recht sich ein bisher besitzloses Stück Land durch Einzäunen, Errichtung einer Heimstätte und Bebauung anzueignen.

Ähnlich wie dieses Landnahmerecht gestaltet sich Raymond zufolge die Eigentumsverteilung im unbegrenzten und schwachbesiedelten Raum der möglichen Softwareprojekte. In der Praxis der Open-Source-Entwicklung haben sich informelle Sitten zur Regelung des Eigentums herausgebildet. Auf dieser informellen Ebene ist es durchaus festgelegt, wer Software ändern darf, und wer das Programm verteilen darf.

Der Eigentümer eines Software-Projektes ist demnach die Person (oder die Personen), die von der Hacker-Community als solcher anerkannt ist. Er hat das exklusive Recht, veränderte Versionen zu verteilen. Es gibt drei Wege, das Eigentumsrecht an einem Projekt zu erwerben:

- die Gründung eines neuen Projektes
- die Übertragung des Eigentums von dem früheren Besitzer. Solche Übertragungen werden meist weithin bekannt gemacht im Usenet, auf Mailing-Listen und/oder in einschlägigen Newsforen<sup>c</sup>.
- die Übernahme eines stillgelegten, verwaisten oder nicht mehr bearbeiteten Projektes. Auch diese Übernahme muß entweder vom vorherigen Besitzer gebilligt werden oder so bekannt gemacht werden, daß der vorherige Besitzer Einspruch erheben kann.

Das Eigentum und die Ausübung der damit verbundenen Rechte der Veränderung und Verteilung der offiziellen Versionen kann folgendermaßen organisiert sein:

- ein einzelner Eigentümer (viele der kleineren Projekte)
- ein übergeordneter »wohlwollender Diktator« und mehrere Co-Eigentümer für einzelne Bereiche (Beispiel Linuxkernel)

<sup>1</sup> Im Regelfall sind die Programme kostenlos aus dem Internet downloadbar. Für CD oder DVD-Versionen wird meist eine geringe Gebühr erhoben, die etwa den Kopierkosten entspricht. Die gängigen im Handel zu findenden *Distributionen*, das sind vorkonfigurierte Softwaresammlungen, sind meist etwas teurer, da meist ein umfangreiches Handbuch oder Installationssupport inbegriffen sind.

<sup>a</sup><http://www.opensource.org/osd.html>

<sup>b</sup><http://catb.org/~esr/writings/cathedral-bazaar/homesteading/>

<sup>c</sup><http://slashdot.org>

- ein Komitee, in dem Entscheidungen nach dem Mehrheitsprinzip oder Einstimmigkeitsprinzip gefällt werden (Webserver Apache)
- »rotierende« Diktatur: die Kontrolle wird periodisch an ein Mitglied des Entwicklerstabs übertragen (Perlprojekt)

Es gibt innerhalb der Hacker-Community einige stillschweigend akzeptierte Tabus, die diese Praxis des Eigentums festigen helfen:

- So existiert starker Widerstand gegen die Aufspaltung (*forking*) eines Software-Projekts.
- Geächtet ist auch die Verteilung einer geänderten Version eines Programms ohne die Zustimmung der »Eigentümer«. Ausgenommen sind geringfügige Anpassungen an eine andere Hardwarearchitektur oder an die Besonderheiten einer speziellen Distribution (wobei auch hierbei normalerweise der Eigentümer auf dem Laufenden gehalten wird).
- Niemals wird der Name eines Entwicklers aus der »Projektgeschichte« oder aus der Dokumentation der Änderungen (*Changes*) ohne dessen ausdrückliche Zustimmung entfernt.

Diese Tabus machen deutlich, daß es in der OS-Bewegung vor allem um Reputation innerhalb der Gruppe geht. Dieser soziale Status wird nicht durch *Anhäufung* von knappen Gütern oder Zeichen erreicht, sondern durch Geschenke an die Community als Ganzes. Das führt zur Ökonomie der OS-Bewegung.

### Parentese: Kapitalismus und Open Source

Softwareentwicklung geschieht nicht im luftleeren Raum, sondern innerhalb der kapitalistischen Produktion, ja im Zentrum des Kapitalismus, dort wo der Umbau hin zu einer rechnergestützten Verwertung stattfindet. Hacker sind in diesen Prozeß eingebunden und bestreiten ihren Lebensunterhalt meist innerhalb dieser ökonomischen Mechanismen. Sie sind (oder sehen sich als) die Elite unter den Programmierern.

Die Entwicklung der Open-Source-Software erfolgt in den Freiräumen, die Schule, Universität, Forschungsinstitute oder Unternehmen ihren hochqualifizierten Beschäftigten gewähren. Ökonomisch ruht die OS-Community auf dem Unterbau der Markt-Ökonomie. Das wird gerade in Krisenzeiten<sup>a</sup>, in denen auch die Mittel der OS-Bewegung schrumpfen, deutlich. Diese Basis bleibt aber aus dem Diskurs der OS-Bewegung ausgeschlossen. Wenn überhaupt, wird das Verhältnis zu Kapital und Marktwirtschaft eher unkritisch betrachtet.

Darüber kann auch die Fehde um die Kommerzialisierung der Open-Source-Bewegung zwischen Richard M. Stallman und Eric S. Raymond nicht hinwegtäuschen. Stallman<sup>b</sup> geht es mit seinem Hippie-Idealismus um den Kampf gegen die marktbeherrschenden Softwarekonzerne im Sinne der bürgerlichen Freiheit, nicht um eine grundsätzliche Infragestellung des Systems. Raymond dagegen sieht im Markt die beste aller möglichen Organisationsformen der Gesellschaft. Eigentum bezeichnet er in einem kruden Biologismus als instinktähnliches Verhalten: »Eigentum ist eine Ableitung von tierischem Revierverhalten, das sich zur Reduzierung von Gewalt innerhalb einer Art entwickelt hat.«<sup>2</sup>

Auch der sozialromantische Idealismus der Oekonux-Bewegung<sup>c</sup>, der durch die Verbreitung freier Software einen evolutionären Umbau der Marktwirtschaft hin zu einer eigentumslosen Gesellschaft

<sup>2</sup>»Property is an abstraction of animal territoriality, which evolved as a way of reducing intraspecies violence.«

<sup>a</sup><http://news.com.com/2100-1001\3-276056.html>

<sup>b</sup><http://www.stallman.org>

<sup>c</sup><http://www.oekonux.org>

propagiert, unterstützt die These von der ökonomischen Blindheit der Bewegung, weil die vorhandenen ökonomischen und politischen Machtverteilungen einfach außer Acht gelassen werden.

### Die Ökonomie der OS-Bewegung

Festzuhalten bleibt jedoch: Auf der kapitalistischen Basis hat sich eine anders geartete Ökonomie entwickelt. Für eine Analyse der inneren Mechanismen der Open-Source-Bewegung kann eine Analogie helfen: Nämlich die zur Geschenkökonomie der archaischen Gesellschaften, wie sie zum Beispiel von Marcel Mauss oder Pierre Bourdieu untersucht und beschrieben wird.

In diesen Gesellschaften ist ein bestimmender Faktor des Verhaltens die eigene Ehre, der eigene Name, das Ansehen. Die Anhäufung von solchem *symbolischen Kapital* besteht in dem Glauben oder Kredit der Gemeinschaft. Es kann in einem solchen System durchaus rational sein, ökonomisch irrational zu handeln. Das erstaunt viele Rational-Choice-Ökonomen. Sie suchen nach eigennützi- gen und geldwerten Motivationen, und finden keine.

Wichtig für die Gewinnung von symbolischem Kapital, von Reputation, ist es, jeden Anschein von Eigennutz zu vermeiden und sich den Werten der Gemeinschaft strikt konform zu verhalten.

Die Hackerkultur könnte demnach als eine Gabentauschkultur gefaßt werden. Das meiste Ansehen genießt derjenige, der der Gemeinschaft die größten Geschenke bereitet. Nicht umsonst ist Linus Torvalds, der verantwortliche Entwickler des bisher einzigen OS-Betriebssystems, der unumstrittene Star der Bewegung. Je besser die Reputation des Einzelnen, je höher seine Anerkennung in der Community, desto größer ist die Chance, Aufmerksamkeit und Kooperation in Form von Programmcode seitens anderer zu bekommen, die das eigene symbolische Kapital vermehren (auch das des Codelieferanten, aber nicht in dem Ausmaß).

Unter diesen Vorzeichen machen auch die oben aufgezählten 3 Tabus Sinn:

- Die Aufspaltung eines Projektes (*forking*) setzt die Beiträge einem Reputationsrisiko aus.
- Ohne Zustimmung des Eigentümers veränderte Versionen können eine gute Reputation zerstören.
- Das Löschen von Namen aus den Projektdateien ist das größte Verbrechen. Jemand wird seiner/ihrer Reputation beraubt. Sein Geschenk an die Community wird unter anderem Namen verteilt.

Gabentausch ist nach Bourdieu die einzig akzeptable Form der (Waren-)Zirkulation in Gesellschaften, die ihre »wahre Lebensgrundlage« leugnen. Das trifft in gewissem Maß auch auf die Open-Source-Bewegung zu. Derartige SoftwareEntwicklung findet in den Freiräumen statt, die (hochqualifizierte) Programmierer neben ihrer Arbeits- oder Ausbildungszeit haben. Also normalerweise außerhalb des regulären Ausbildungs- oder Beschäftigungsverhältnisses. Dieser »Brotjob« ist jedoch die »wahre Lebensgrundlage«. Das ist der »blinde Fleck« der Ökonomie im Diskurs der Open-Source-Bewegung. Das eigene Verhältnis zur kapitalistischen (Software-)Produktion wird nicht thematisiert. Aus dieser Basis wird jedoch ein nicht unbeträchtlicher Teil der Motivation der Bewegung gewonnen. Das Programmieren für die Community geschieht *just for fun*. So lautet auch der Titel der Autobiographie des maßgebenden Linuxentwicklers Torvalds. In der Open-Source-Software ist der Entwickler unabhängig von zugewiesenen Projekten, Zeitplänen und Deadlines. Er sucht sich sein Projekt selbst aus, meist anhand eigener Schwierigkeiten oder Vorlieben (»*Every good work of Software starts by scratching a developer's personal itch*«, wie Raymond schreibt) und trifft (als »Eigentümer«) wichtige Entscheidungen selbst. Open Source ermöglicht den Programmierern also einen viel selbstbestimmteren Arbeitsprozeß, als es die lästige Lohnarbeit erlaubt.

Die Analogie mit der Gabentauschkultur hat somit auch ihre Mängel. Sie erklärt nur einen Teil der Motivation, da sie die ökonomische Basis vernachlässigt. Sie erklärt jedoch das Reputations-Spiel recht gut und sie läßt einige weitere Hypothesen zu dieser Form von Online-Community zu:

### Macht

Die Analyse des symbolischen Kapitals führt direkt zur Analyse der Machtstrukturen innerhalb der OS-Bewegung. Ein Machtstreben der Projekt-Eigentümer wird von den Protagonisten immer ausgeschlossen. Dabei wird Macht aber eingeschränkt auf direkten Zwang. Daß sich auch in dieser auf Freiwilligkeit beruhenden Community Machtstrukturen etablieren, ist nach dem bisher Gesagten offensichtlich. Nicht in Form einer direkten Erpressung von Programmcode, sondern subtiler über den Gabenmechanismus.

Eine Gabe bringt den Nehmenden, also den Nutzer des Programms, in die Schuld des Gebers. Es muß eine Gegengabe erfolgen, sei es in Form eines *bugreports* (dem Schildern eines Programmfehlers) oder gar eines *patches* (der Behebung eines Fehlers). Das funktioniert keineswegs bei allen Benutzern des Programms, aber doch bei hinreichend vielen, um das Weiterbestehen vieler Projekte und ein enormes Wachstum der Community zu ermöglichen (mehr als 30 000 Projekte werden zur Zeit auf einem zentralen Server<sup>a</sup> gehostet).

Die unterschweligen Eigentumsverhältnisse garantieren so den zentralen Personen eine Sicherung und Vermehrung ihres symbolischen Kapitals und ihrer symbolischen Macht. Die Macht liegt zuerst in der Entscheidungsgewalt über die weitere Entwicklung des Programms: Welche Funktionen werden neu aufgenommen (oder gestrichen)? Wo wird das Design verändert? etc. Diese Macht unterliegt, wie in den archaischen Gesellschaft, vielfältigen Restriktionen und Kontrollen durch die Community. Bei ›falschen‹ Entscheidungen besteht immer die Gefahr der Aufspaltung des Projekts. Letztendlich bleibt sie immer abhängig von der Anerkennung der Community. Die symbolische Macht erfordert deshalb aber auch einen großen persönlichen Einsatz: Entscheidungen müssen plausibel kommuniziert werden, Nutzer und Entwickler müssen ›bei der Stange‹ gehalten werden. Symbolische Macht bestimmt die Wirkung eines Beitrages zum Diskurs der OS-Bewegung. So wird Raymond in Newsgruppen als ›Trittbrettfahrer‹ der Bewegung bezeichnet, weil dessen Projekt fetchmail mit den großen Projekten Linux, Emacs, oder gcc nicht mithalten könne.

Derartige symbolische Macht, die bisher stark an Personen gebunden ist, wird sich langsam institutionalisieren. Vor allem Positionen in den großen Projekten bekommen zunehmend eine von Personen unabhängiges Machtpotential (zum Beispiel die Position des Kernel-Co-developers).

Die Macht und Reputation in der Open-Source-Bewegung kann natürlich auch Rückwirkungen auf die eigene ökonomische Basis haben und sich in einer besseren Position oder Bezahlung niederschlagen. Die Stars der Bewegung erhalten auch Jobs, in denen sie ihrem ›Hobby‹ Softwareprogrammierung uneingeschränkt nachgehen können (»*So in a sense I do get my pizzas paid for by Linux indirectly*« Torvalds<sup>b</sup>).

Und schließlich ist die Macht der kapitalisierten Teile der Open-Source-Bewegung nicht zu vergessen. Open-Source-Unternehmen üben inzwischen Kontrolle über einen guten Teil der Entwicklung aus. Das Unternehmen VA Software betreibt und kontrolliert den zentralen Server<sup>c</sup> für Open-

<sup>a</sup><http://sourceforge.net>

<sup>b</sup><http://www.firstmonday.dk/issues/issue3\3/torvalds/>

<sup>c</sup><http://www.vasoftware.com/products/>

Source-Projekte (was schon zu einer Warnung<sup>a</sup> der Free Software Foundation führte) und das zentrale Nachrichtenbrett<sup>b</sup> der Community. Firmen wie RedHat<sup>c</sup> oder SuSE<sup>d</sup> beschäftigen eine zunehmende Anzahl Open-Source-Entwickler und nehmen so erheblichen Einfluß auf Richtung und Ziele der Open-Source-Softwareentwicklung. Sie bringen inzwischen sogar ihre eigenen Kernelversionen heraus. Gleichzeitig nehmen die Investitionen (>Partnerschaften<) von großen Firmen wie IBM oder Oracle immer mehr zu. Hier öffnet sich ein weites Feld von subtilen Einfluß- und Kontrollmöglichkeiten, in dem die symbolischen Machtstrukturen in der Community eine wichtige Rolle spielen. Ein Feld, das in der Hacker-Community bisher nicht beachtet wird, das jedoch für die zukünftige Entwicklung maßgebend sein wird.

### Ausblick

Bemerkenswert an der Besiedlung des Internetaums ist die Tendenz zu »stammesorganisatorischen Verhaltensweisen« (McLuhan), die gerade auf die archaischen Mechanismen der Geschenkökonomie und der Ehre zurückgreifen. (Das ist keineswegs auf die Open-Source-Gemeinde beschränkt. MUDs, Newsgruppen, IRCs etc. weisen ebenfalls Analogien dazu auf. Direkt greifbar ist dieses Phänomen in den >Bewertungen< von Käufer oder Verkäufer der Online-Auktionen.)

Die Open-Source-Bewegung hat in Zeiten starken ökonomischen Wachstums (zumindest im Bereich der Informations- und Kommunikations-Technologien) stark an Verbreitung gewonnen. Die Stärke und Dauerhaftigkeit der Open-Source-Bewegung wird erst in den anbrechenden Krisenzeiten deutlich werden. Wenn die bisher recht freizügig gewährte Unterstützung der Industrie (IBM, Compaq, diverse Linuxfirmen) sich minimiert oder konkretere Gegenleistungen dafür gefordert werden, wenn der Einfluß von kapitalkräftigen Unternehmen auf die weitere Entwicklung weiter wächst, wenn die gewährten Freiräume für die Programmierung freier Software kleiner werden und wenn ein Teil der Open-Source-Gemeinde sich durch sozialen Abstieg bedroht sieht.

Ob der Raum der freien Softwareentwicklung vom Kapitalismus kolonialisert wird, ist nicht die Frage, sondern nur wann und in welcher Form. Wie sich dabei die Open-Source-Bewegung verändert, wird spannend zu beobachten sein. Daß der Kapitalismus sich dabei *grundlegend* verändern wird, ist nach den bisherigen Erfahrungen mit dieser Form der Ökonomie zu bezweifeln.

### Literaturhinweise

Bourdieu, Pierre: *Sozialer Sinn. Kritik der theoretischen Vernunft* Frankfurt am Main: Suhrkamp 1987.

McLuhan, Marshall: *Die magischen Kanäle. Understanding media*. Dresden: Verlag der Kunst 1995.

Raymond, Eric S. (2001). Homesteading the Noosphere. In: Raymond, Eric S. (Hrsg.): *The Cathedral and the Bazaar*. Sebastopol: O Reilly 2001, S. 65-112.

<sup>a</sup><http://www.fsfeurope.org/news/article2001-10-20-01.en.html>

<sup>b</sup><http://www.osdn.com/>

<sup>c</sup><http://www.redhat.com>

<sup>d</sup><http://www.suse.de>