SSOAR

Open Access Repository
www.ssoar.info

Kreativität als Norm - zum Erfolg verdammt? Open-Source- Software zwischen sozialer Bewegung und technischer Innovation

Holtgrewe, Ursula

Preprint / Preprint
Sammelwerksbeitrag / collection article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:
SSG Sozialwissenschaften, USB Köln

Empfohlene Zitierung / Suggested Citation:
Holtgrewe, U. (2001). Kreativität als Norm - zum Erfolg verdammt? Open-Source- Software zwischen sozialer Bewegung und technischer Innovation. In J. Allmendinger (Hrsg.), *Gute Gesellschaft? : Verhandlungen des 30. Kongresses der Deutschen Gesellschaft für Soziologie* (S. 399-424). Opladen: Leske u. Budrich. https://nbn-resolving.org/urn:nbn:de:0168-ssoar-122638

Nutzungsbedingungen:
Dieser Text wird unter einer Deposit-Lizenz (Keine Weiterverbreitung - keine Bearbeitung) zur Verfügung gestellt. Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.
Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use:
This document is made available under Deposit Licence (No Redistribution - no modifications). We grant a non-exclusive, non-transferable, individual and limited right to using this document. This document is solely intended for your personal, non-commercial use. All of the copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute or otherwise use the document in public.
By using this particular document, you accept the above-stated conditions of use.

gesis Leibniz-Institut für Sozialwissenschaften

Mitglied der Leibniz-Gemeinschaft

Kreativität als Norm – zum Erfolg verdammt? Open-Source-Software zwischen sozialer Bewegung und technischer Innovation[1]

Ursula Holtgrewe

Erscheint in: Allmendinger, Jutta (Hrsg.): Gute Gesellschaft? Konstruktionen sozialer Ordnungen. Verhandlungen des 30. Kongresses der Deutschen Gesellschaft für Soziologie in Köln 2000. Opladen (Leske&Budrich)

„Allgemein werden ja mit der neuen Ökonomie Gegner und Partner identisch, aber gleichzeitig tun sich auch immer wieder neue soziale Bewegungen auf, die die Arschbacken der Homogenität auseinanderreißen – oder wie soll man sagen?“ (Helmut Höge in der taz vom 4.11.2000: 12)

Der Beitrag geht der Frage nach, was geschieht, wenn in der multimedial vernetzten Wissensgesellschaft Kreativität selbst zur Norm und grundlegenden Anforderung an die sozialen Subjekte wird, während sich die Bewertungsmaßstäbe der Kreativität selbst verändern. Gesellschaftsweit geht es in abnehmendem Maß um soziale Transformation, zunehmend um marktgängige Innovation, wenn das Modell unternehmerischer und an zahlungskräftiger Nachfrage orientierter Innovation sich in soziale Sphären ausbreitet, die herkömmlicherweise anderen Logiken folgen. Dann erscheint der Markt – eine hochspezifische Form gesellschaftlicher Bewertung von Kreativität – als Zielpunkt kreativen Handelns allgemein. Damit jedoch steigt das Risiko, dass gerade jene spannungsvolle Vielfalt von Handlungs- und Erfahrungsräumen sozial eingeebnet und „zum Erfolg verdammt“ wird, aus der soziale Innovationen und Aufbrüche entstehen können. Jedoch macht es Sinn, solche Tendenzen soziologisch nicht einfach als „vollendete Tatsachen“ (Hack 1988) zu betrachten und damit zu verdoppeln, sondern die Aufmerksamkeit für die Vielfalt von Zusammenhängen, Orientierungen, Werten und Institutionen zu schärfen, in denen kreativ und innovativ gehandelt wird – und eben weiterhin von Kreativität in einem auch gesellschaftstheoretisch anspruchsvollen Sinne zu reden.

Deswegen untersuche ich hier ein Beispiel dafür, dass kreatives Handeln nicht nur marktliche Erfolgskriterien in Frage stellt, sondern sich auch die Erhaltung und Erweiterung seiner eigenen Bedingungen zum Ziel setzt. Open-Source- bzw. freie Software markiert ein Feld technischer Innovation, in dem Projekte und EntwicklerInnen-Communities explizit eine „andere“ Ökonomie öffentlicher Güter und freiwilliger Gaben konstituieren. Dabei sind jedoch Verzweigungen entlang der Frage auszumachen, worauf sich kreatives Handeln bezieht: Auf technische Innovation, die gerade wenn sie „erfolgreich“ ist, mehr oder minder direkt unter die Anforderungen und Zwänge marktlichen Erfolgs gerät, oder darüber hinaus auf den Entwurf eines politisch-normativen Gegenmodells zwischen Basisdemokratie und technischer Exzellenz. An zwei Beispielen aus dem Open-Source-Kontext wird deutlich werden, wie die Grenzen zwischen verschiedenen gesellschaftlichen Bewertungsweisen von Kreativität in Prozessen kreativen Handelns erst gezogen, verschoben und verhandelt werden.

1 Für wichtige und nützliche Hinweise und Ideen zu diesem Papier danke ich Sandra Arzbächer, Ingo Bode, Christian Kerst, Peter Sanders, Stephan Voswinkel und Gabriele Wagner. Für Fehler und Ungenauigkeiten ist die Autorin allein verantwortlich.

1. Technische Innovation, Kreativität des Handelns und soziale Bewegung

Die „Entdeckung“ der Kreativität in der Soziologie (Joas 1996), die immer wieder gemacht und vergessen wurde, ist eng mit der expressiven und identitätspolitischen Seite sozialer Bewegungen verbunden. Hier versuche ich, diese Perspektive auf bestimmte Netzwerke informationstechnischer Innovation anzuwenden. Dort spielen solche Orientierungen eine überraschend lebendige Rolle. Damit lassen sich die techniksoziologischen Befunde über die Innovation in Netzwerken ans Motto des Soziologiekongresses 2000 anschließen: Es geht um die normative Deutung und Bewertung kreativen und innovativen Handelns und seiner Produkte. Im Innovationsprozess selber entstehen nicht nur Artefakte, sondern auch *communities*, Identitäten und Werte.

Die Techniksoziologie hat deutlich gemacht, dass sich technische Innovationen im Prozess ihrer Entstehung, Stabilisierung und Durchsetzung durch verschiedenartige Handlungssphären und Netzwerke bewegen, und dass der Aufbau solcher Netzwerke Teil innovativen Handelns ist (Rammert 1997; Weyer 1997; Bender 1999). Damit wird die Heterogenität der beteiligten Institutionen, Akteure und Orientierungen angesprochen, und gerade aus deren Kombination entsteht ja Neues. Die Deutung und Bewertung des Neuen ist in dieser Sicht eingelassen in Innovationsmilieus und -paradigmen. Sie ist etabliert in technischen Leitbildern oder den „mitgebrachten“ professionellen und kulturellen Orientierungen der Innovatoren, aber diese Kulturen und Identitäten werden im selben Prozess reproduziert und transformiert.

Dabei ist Innovation nur in Relation zu gesellschaftlichen Institutionen denkbar, wird durch diese (mit-)konstituiert und setzt sie voraus. Werner Rammert (1997) und der Kunsttheoretiker Boris Groys (1997) haben darauf aufmerksam gemacht, dass zur Anerkennung einer Innovation als solche die Aufnahme in die verschiedenen gesellschaftlichen „Archive“ gehört, die dafür angelegt werden: Museen und Kulturindustrie, aber auch die Institutionen geistigen Eigentums, das Patent- und Urheberrecht. Auch die Kriterien der Bewertung der Innovation sind institutionalisiert: Die Verbreitung und der Erfolg auf dem Markt, aber auch in gewisser Spannung dazu die gebrauchswertorientierten und ästhetischen Maßstäbe technischer oder künstlerischer Exzellenz.

Die Perspektive sozialer Bewegung, mit der ich hier die der Institutionalisierung ergänzen möchte, bezieht sich nun auf die Transformation von Normen, Werten und Institutionen selbst. Soziale Bewegungen setzen sich solche Transformationen nicht einfach als Ziel, sondern nehmen sie in ihrer Praxis vorweg. Sie erzeugen neue soziale Ordnungen selbst (Joas 1996; Barker/Kennedy (Hrsg.) 1996), und kämpfen um die Anerkennung ihrer Praxen (Honneth 1994). In sozialen Bewegungen mit deutlich kulturellem Akzent wird Kreativität selbst zum Thema.[2] Sie erheben Ansprüche auf Selbstverwirklichung und Demokratisierung und klagen soziale Freiräume für kreatives Handeln ein oder „besetzen“ sie. Künstlerische Avantgarden etwa begnügen sich nicht mit der Anerkennung ihrer Erzeugnissse durch die „zuständigen“ Institutionen, sondern stellen deren Wahrnehmungs- und Bewertungskriterien und letztlich sie selbst in Frage. Die Kreativität des Handelns „an der Basis“ wird also nicht nur konstatiert, sondern soll in der Logik sozialer Bewegungen möglichst früh und möglichst breit in Prozesse des sozialen Wandels und der gesellschaftlichen Innovation eingespeist werden, um nicht nur das Potential für Innovationen zu erhöhen, sondern diese zu gestalten und die Innovationen selbst zu demokratisieren.

Aktuell scheint es, dass sich diese Verbindung von Kreativität und sozialer Transformation aufgelöst hat, obwohl – oder weil – von Innovation allerorten die Rede ist und Kreativität als Norm in der Tat sozial diffundiert ist. Kulturell und medial vermittelt hat sich das Leitbild dieser organisationell und gesamtgesellschaftlich geforderten Kreativität jedoch verändert. Sie wird in der Wissensgesellschaft verstanden als in erster Linie unternehmerische (Baecker 1997; Voß/Pongratz 1998; Moldaschl/Sauer 2000). Damit erfährt die Kreativität als Norm

2 Zu denken ist an die Traditionen der Arbeiterkultur, aber auch an Frauenbewegung, MigrantInnengruppen und (Jugend-)Subkulturen. Mit Schwendter (1973/1981) kann man etwas altmodisch von progressiven Subkulturen sprechen, die sich nicht nur von der Gesamtgesellschaft unterscheiden, sondern diese zu transformieren und ihre eigenen Aktionsfelder zu erweitern suchen.

eine gewichtige Transformation. Es ist weniger der schöpferische Prozess in offene Möglichkeitsräume hinein, auf den es ankommt, als der Erfolg auf dem Markt. Damit geht es zunächst einmal nicht mehr um Aufbrüche über geltende Normen und Bewertungen hinaus, sondern gerade um eine spezifische Form der Bedienung von zahlungskräftigen Erwartungen. Nicht nur technische, sondern auch soziale und kulturelle Innovationen werden dann mit dem Markt kurzgeschlossen. Soziale Räume und Zeiten werden ausgeblendet und entwertet, in denen alternative, nicht marktgängige soziale und kulturelle Neuerungen ausprobiert oder gar Alternativen der Kooperation entwickelt werden könnten.

Aus eben diesem Grund erscheint es sinnvoll, die Orientierungen und Dispositionen sozialer Bewegungen nicht einfach abzuschreiben, sondern den vielfältigen, politisch und sozial transformativen kulturellen Beständen und Entwürfen nachzugehen, die sich auch und gerade in Feldern technischer Innovation finden. Der Punkt, an dem sich die Relevanz der Frage nach der sozialen Bewegung in Innovationsnetzwerken bestimmen lässt, ist dann, auf *welche* Bewertungskriterien sich das kreative Handeln bezieht. Erfolg nach gesamtgesellschaftlich etablierten Kriterien, am Markt oder in Gestalt öffentlicher Aufmerksamkeit kann zwar die kreative Leistung bestätigen, aber bestätigt gleichzeitig die etablierten Kriterien und die Institutionen, die diese verkörpern. Darum stehen solche Erfolge vom Standpunkt der Kritik und Transformation aus auch wieder unter Verdacht: Hat man nicht doch gesellschaftliche Erwartungen in stärkerem Maß bedient und vollzogen, als sich dazu schöpferisch verhalten? Das Hochhalten der Grundlagenkritik und puristisch-avantgardistischen Praxis gegenüber den Erfolgskriterien dagegen setzt sich dem Verdacht der Spinnerei und Wirkungslosigkeit aus.

Das Verhältnis von Kreativität und Norm ist also zweischneidig, und immer wieder erst herzustellen. Die Fragen von Anpassung und Autonomie, Subkultur und Ausverkauf, Exzellenz und Erfolg sind aufeinander bezogene, klassische Differenzierungs- und Konfliktlinien sozialer und kultureller Bewegungen und Projekte – aber sie werden auch *in* den einzelnen Projekten und Netzwerken und mitunter quer durch die Individuen selbst ausgehandelt.

2. Open-Source-Software: Hochqualifizierte Nicht-Arbeit

Open Source oder „freie“ Software – zu den feinen Unterschieden s. Abschnitt 4 – ist ein Beispiel, in dem Technikentwicklung außerhalb von Organisationen und auch von bezahlter Arbeit stattfindet und in dem dieser Aspekt zentral wird.[3] In der Informations- und Kommunikationstechnik haben die Praxen von Hobbbyisten Tradition seit den Radiobastlern der 20er und 30er Jahre. Auch die Entwicklung des PC wurde in der Frühzeit von BastlerInnen betrieben, die dazu Clubs, Fachzeitschriften und schließlich Firmen gründeten und damit durchaus auch alternativkulturelle Ziele verfolgten, Technik zu demokratisieren und allgemein zugänglich zu machen (Schmid 1997).

Wenn also im folgenden von Hackern jeglichen Geschlechts die Rede ist, ist damit die Fortsetzung dieser Traditionen gemeint: „Computerfreaks“ im Unterschied zu professionellen EntwicklerInnen,[4] für die das Programmieren einen weit gehend intrinsischen, expressiven und Identität stiftenden Charakter hat. (Gemeint sind also nicht die BetreiberInnen illegaler Einbrüche in Datennetze und Computersysteme.) Zentrale normative Orientierungen dieser *communities* sind technische Exzellenz, Freiheit der Information, der intrinsische Wert des Programmierens, eine tendenziell anarchistische oder libertäre Ablehnung von Hierarchien,

3 Einen guten Überblick gibt Grassmuck 2000, eine kritisch-sympathisierende Materialsammlung, die just-in-time gepflegt wird, hat Bezroukov (2000) erstellt, s. vom selben Autor auch www.softpanorama.org/oss/bla_faq und www.softpanorama.org/People/. Die wichtigsten programmatischen Positionen versammeln DiBona/Ockman/Stone (Hrsg.) 1999. Der vorliegende Beitrag basiert auf einer explorativen Dokumenten- und Literaturauswertung, die sich auf Internetdokumente, Diskussionsforen und Selbstdarstellungen der Szene stützt. Informativ ist die Internet-Dokumentation des Kongresses „Wizards of OS“ am 16./17. Juli 1999 in Berlin unter www.mikro.org/Events/OS/. Websites von Projekten und Organisationen sind www.gnu.org, www.opensource.org, www.osdn.com, www.linux.com und sehr viele mehr. Als Nachrichten- und Informationsdienste und Diskussionsforen sind zu nennen www.slashdot.com, www.newsforge.com, www.heise.de/tp, www.linux.com/linuxgram, www.linuxtoday.com. Usenet-Newsgroups, die sich mit Open-Source-Programmatik befassen, sind comp.os.linux.advocacy und gnu.misc.discuss.

4 Wobei dies durchaus dieselben Personen sein können, vgl. die Beiträge in Krafft/Ortmann (Hrsg.) 1988 und Strübing 1993.

kurz „eine kreative Lebenseinstellung“(so der Chaos Computer Club)[5], die sich in der Tat auf die Sicherstellung der Bedingungen kreativen Handelns bezieht.

Diese Szenen sind schwerlich auf einen Begriff zu bringen: Sie haben Züge einer professionellen Kultur, einer kulturellen Avantgarde, aber vielleicht noch eher einer *community of practice* (Wenger 1998; Moon/Sproull 2000), die sich um das Programmieren als bestimmte kreative Praxis gruppiert. Charakteristika einer sozialen Bewegung sehe ich dort, wo die Akteure die skizzierten Orientierungen sozial zu generalisieren suchen und sie zum Kriterium der Bewertung und Transformation gesellschaftlicher Verhältnisse und Institutionen machen.

Mit *Open Source* befinden wir uns hier überwiegend in der für normale PC-NutzerInnen fremden und seltsamen Welt des Betriebssystems Unix und seiner Verwandten. Diese Betriebssysteme sind im Unterschied zu DOS oder Windows so aufgebaut, dass sie eine Vielzahl von einzelnen Werkzeugen zum Aufbau einer Arbeitsumgebung bereit stellen. Das ist deswegen wichtig, weil diese modulare Architektur das dezentral verteilte Arbeiten an kleineren Programmen und Programmbestandteilen wesentlich erleichtert (Moon/Sproull 2000). Herkömmlicherweise waren BenutzerInnen meist ihrerseits EntwicklerInnen, und der Übergang von „Arbeitsmitteln“ und „Arbeitsgegenständen“ ist fließend.

Bei Open-Source-Projekten handelt es sich um nichtkommerzielle Software-Entwicklungsprojekte, in denen eine Vielzahl Beteiligter – von einigen Dutzend bis zu Hunderten oder sogar einigen tausend SoftwareentwicklerInnen – Programmbestandteile und -module schreiben, testen, verbessern und dokumentieren. Mit den Möglichkeiten elektronischer Vernetzung können solche Weiterentwicklungen fast ohne Kosten ins Netz gestellt und anderen weitergereicht werden. Räumlich und zeitlich getrenntes Arbeiten an dem Programm(paket) wird möglich. Der Anspruch ist ein dezidiert öffentlicher: Die Einstiegshürden sind niedrig, alle sind zur Mitarbeit eingeladen, die einen qualifizierten Beitrag leisten können. (Die schwierige und oft wenig explizite Hürde macht natürlich aus, was als qualifizierter Beitrag angesehen wird.) Diskussion findet weitgehend über öffentliche Mailinglisten statt. Die Entscheidung, welche Beiträge in neue Softwareversionen aufgenommen werden, wird jedoch meist ziemlich zentral getroffen. Das bekannteste Open-Source-Beispiel ist wohl das Betriebssystem Linux, das sich in den letzten Jahren auch auf private PCs verbreitet hat. Aber auch wer das WWW nutzt oder e-mail verschickt, greift auf freie Software zurück, unter der – für die Normalnutzerin meist hinter den Kulissen – Web- und Mailserver laufen.

Was heißt nun *Open Source*? Wörtlich geht es um die Zugänglichkeit des Quellcode eines Programms. Programme werden im Allgemeinen in höheren Programmiersprachen geschrieben – das ist der Quellcode – und von dort aus in Maschinensprache übersetzt (kompiliert). Den Quellcode braucht man dann, wenn man ein Programm auf seine Funktionsweise untersuchen, umbauen, verbessern oder weiterentwickeln möchte. Eben dies schließt proprietäre Software in den meisten Fällen aus, denn dieses Recht behält sich der Hersteller vor.[6] Wenn man ein Programmpaket kauft, findet man auf der CD-ROM im Allgemeinen die binäre, also kompilierte Version, die zwar den Rechner veranlasst, das zu tun was er soll, aber aus der nicht zu entnehmen ist, was die Programmiererin geschrieben hat. Den Quellcode von dort zu rekonstruieren, ist extrem aufwändig bis unmöglich und wird zudem durch die meisten Benutzungslizenzen verboten.

Das war nicht immer so. Software als eigenständige Ware gab es bis in die 70er Jahre hinein nicht. Hardwarehersteller lieferten sie frei mit oder Anwender schrieben ihre Programme selbst, und was an Universitäten entwickelt wurde, wurde frei verbreitet (Grassmuck 2000). Auch das Betriebssystem Unix und die Programmiersprache C waren zunächst interne Entwicklungen aus den Bell Labs, dem F&E-Labor des zwar privaten, aber staatlich regulierten us-amerikanischen Telefonmonopolisten AT&T –mit dem ausdrücklichen Ziel der Autoren, nicht nur eine gute Programmierumgebung zu entwickeln, sondern „a system around which a fellowship could form“ (Dennis Ritchie, zit. nach Grassmuck 2000: 3). Insofern ist festzuhalten, dass es sich bei *Open Source* nicht einfach um einen Aufbruch aus den konventionellen

5 Vgl. Levy 1984; Kuwabara 2000 und den Chaos Computer Club (http://www.ccc.de/Hackerethik.html).

6 Weder freie Verbreitung noch offener Quellcode definieren die Open-Source-Szene erschöpfend: Es gibt auch kommerzielle Software, insbesondere Programmbibliotheken, die mit dem Quellcode geliefert werden, und die kostenlos vertriebene Free- und Shareware aus dem PC-Bereich wird meist als Objektcode verteilt.

Welten kommerzieller Softwareentwicklung handelt. Vielmehr setzen die Open-Source-*communities* technisch wie sozial Traditionen des Gemeinwohls und des Gebrauchswerts öffentlicher Güter fort, die aus der Wissenschaft und der Logik öffentlicher Infrastrukturen stammen, und die auch für die Entwicklung des Internet konstitutiv waren (vgl. Werle 2000). Eben diese Traditionen suchen sie zu verteidigen und auszubauen.

Open-Source-Projekte haben ihre Gründungsgeschichten und -mythen. Meist entstehen sie aus einer Idee oder einem Problem eines Hackers: „Every good work of software starts by scratching a developer's personal itch" (Raymond 1998: 3). Das kann etwa eine fehlende Funktionalität eines Programms sein, das man benutzt; der Versuch, es auf eine andere Hardware zu portieren oder ähnliches. Zur Lösung eines solchen Problems schreibt jemand sich ein Programm, stellt es ins Netz und bittet um Beiträge, Tests, Korrekturen. Wenn das Vorhaben ein „plausibles Versprechen" (Raymond 1998) auf Funktionalität oder auf weitere interessante Probleme enthält, besteht die Chance, dass sich MitentwicklerInnen finden und das ganze die Form eines Projekts annimmt, in dem entwickelt, implementiert und getestet wird, Programmfehler entdeckt und neue Features eingebaut werden. Dabei sind die Grenzen zwischen NutzerInnen und EntwicklerInnen zunächst ausgesprochen fließend:

„Users are wonderful things to have, and not just because they demonstrate that you're serving a need, that you've done something right. Properly cultivated, they can become co-developers" (Raymond 1998: 5).[7]

Bei größeren Projekten bilden sich meist – über die GründerIn des Projekts oder deren NachfolgerIn hinaus – Kernteams, informell emergierende Arbeitsteilungen und Zuständigkeiten. Bei den Verantwortlichen bleibt die Auswahl und Aufnahme der Beiträge in neue Versionen. Es gibt jedoch Hinweise, dass die Beiträge zu Open-Source-Projekten recht ungleich verteilt sind, und diese „zentraler" an den Kern-AutorInnen hängen als es die verbreitete Selbstbeschreibung eines „Basars" (Raymond 1998) dezentraler Entwicklung vorsieht. Ghosh und Prakash (2000) fanden bei ihrer computergestützten Auswertung von AutorInnen und Codezeilen, dass die Verteilung der Beiträge hoch ungleichgewichtig ist: In den von ihnen ausgewerteten Projekten hatten die aktivsten 10% der AutorInnen 72,3% der Codezeilen beigetragen.[8] Unter den rund 13.000 AutorInnen der Linux-Kern-Mailingliste (s. www.tux.org/lkml), die zwischen Juni 1995 und April 2000 insgesamt 175.000 Betiträge verfassten, zeichnen 2% für über die Hälfte der Beiträge verantwortlich (Moon/Sproull 2000: 13).

3. *Open Source*: Anreize, Interessen und Motive

Wie funktionieren nun Open-Source-Projekte, was motiviert die EntwicklerInnen und wie werden ihre Beiträge koordiniert? Sozialwissenschaftlich gibt es dazu vergleichsweise wenig[9], und die vorliegenden Untersuchungen sind nahe an der Selbstbeobachtung der Szene. Diese hat ihre Theoretiker und Programmatiker (Eric S. Raymond, Richard Stallman, Rishab Aiyer Ghosh, Nikolai Bezroukov, und in Deutschland Volker Grassmuck und Stefan Meretz), die auch auf sozialwissenschaftliche Kategorien zurückgreifen. Es finden sich Erklärungen entlang einer Geschenk- oder Reputationsökonomie (Raymond 1998; 1999, aber auch sein Kritiker Bezroukov 1999a und b), Überlegungen zur Ökonomie öffentlicher Güter (Ghosh 1999; Kuwabara 2000), aber auch entlang intrinsischer und normativer Motive (Grassmuck 2000) bis hin zu einem Selbstverständnis als quasi soziale Bewegung (Stallman 1999; FifF 1999, und mit einem kapitalismuskritischen Akzent Meretz 2000 und die Gruppe Oekonux, www.oekonux.de).

---

7 Insofern ist es einigermaßen ironisch, wenn die Managementgurus der Kundenorientierung ausgerechnet die Firma Microsoft, den stilisierten Gegenspieler der freien Software, als Paradebeispiel für erfolgreiche Motivation von Kunden zum Softwaretesten hervorheben (Prahalad/Ramaswamy 2000).

8 Das Ergebnis wird allerdings dadurch beeinflusst, dass die Beiträge der Free Software Foundation und auch die aus bestimmten Firmen keine einzelnen AutorInnen ausweisen.

9 Zu nennen sind einige Befunde aus dem „Projekt Kulturraum Internet" (Helmers u. a. 1998), eine soziologische Abschlussarbeit von der Cornell University (Kuwabara 2000) und als kulturanthropologischer Beitrag Risan o. J. , aus einer organisationstheoretischen Sicht Moon/Sproull 2000.

3.1 Kollektives Handeln zwischen Reputation und Intrinsik

Es bietet sich also an, die verschiedenen Gedankengänge um die verschiedenen Aspekte kollektiven Handelns zu gruppieren.[10] Olson zufolge (1968) wird die Erstellung öffentlicher Güter umso unwahrscheinlicher, je leichter es ist, diese zu nutzen ohne selber Beiträge dazu zu leisten (also Trittbrett zu fahren). Um Akteure zu Beiträgen zu motivieren, braucht es demnach selektive Anreize, die nur den Aktiven zu Gute kommen. Kollektive Güter ohne selektive Anreize zu erstellen, ist dann möglich, wenn die Akteure über die Beiträge anderer informiert sind, und wenn bestimmte Akteure an dem kollektiven Gut ein so hohes Interesse haben, dass sie eine „kritische Masse" von Vorleistungen erbringen, die andere dann zu Beiträgen motiviert (Marwell/Oliver 1993, vgl. Bode 1997; Kuwabara 2000). Geht es um soziale, kulturelle und eben technische Innovationen, so kann die Institution des Autors oder der Erfinderin den Impuls zur Erstellung öffentlicher Güter geben. Hier bringt die Zuschreibung der Urheberschaft oder Gründung eines Projekts spezifische Anerkennungs- und Reputationschancen für die InitiatorInnen.

Kollektives Handeln im Sinne sozialer Bewegungen weist über diese gewissermaßen instrumentellen Motive hinaus. Dort werden vielfach nicht nur Interessen, sondern auch Verletzungen von Wertüberzeugungen oder Anerkennungsansprüchen zum Ausgangspunkt, soziale Verhältnisse so zu verändern, dass diese eingelöst werden. Dazu wiederum schaffen soziale Bewegungen sich Gemeinschaften, die die Geltung dieser Ansprüche antizipieren (Honneth 1994; Holtgrewe/Voswinkel/Wagner 2000; Holtgrewe 2000).

Der Open-Source-Programmatiker Eric S. Raymond argumentiert in seinen Aufsätzen „The Cathedral and the Bazaar" (1998) und „Homesteading the Noosphere" (1999) entlang der selektiven Anreize, dass *Open Source* nach den Regeln einer Geschenkökonomie funktioniert, in der Gaben gegen Reputation eingetauscht werden.[11] Die impliziten Regeln der Entwicklergemeinden stellen demnach sicher, dass Reputation gerecht verteilt wird und dass die Informationen darüber ohne Störungen zirkulieren können. Entsprechend gestalten sich – so Raymond – die Führungs- und Autoritätsstrukturen: Reputation wird erworben, indem anerkannte Autoritäten – die GründerInnen oder Verantwortlichen eines Projekts – freiwillig erbrachte Beiträge bewerten, sie in neue Versionen aufnehmen oder ablehnen. Bezogen ist diese personengebundene Autorität und Reputation freilich auf andere, gewissermaßen professionelle Normen, die vorsehen, dass es auf dem Reputationsmarkt letztlich nicht um das Ego, sondern um die Sache, die funktionierenden und guten Programme gilt. Reputation funktioniert also – wie im richtigen Leben – als Zustand, der nicht direkt, sondern wesentlich als Nebenfolge angestrebt werden kann (Elster 1987: 141ff., vgl. Voswinkel 1999). Aus diesem Grund wird von den Autoritäten Bescheidenheit und Understatement im Auftreten erwartet, was ihnen wiederum Reputation als vertrauenswürdige Treuhänder der Reputation ihrer MitentwicklerInnen einbringt:

„Interestingly enough, you will quickly find that if you are completely and self-deprecatingly truthful about how much you owe other people, the world at large will treat you like you did every bit of the invention yourself and are just being becomingly modest about your innate genius" (Raymond 1998: 11).

Während Raymond die Reputationsökonomie als markt-analoge Lösung für vielfältige Probleme auch des kommerziellen Software-Engineering betrachtet, macht Bezroukov (1999a und b) auf die möglichen Disfunktionalitäten aufmerksam. In Analogie zu anderen Reputationsökonomien, insbesondere der Wissenschaft, weist er auf den inhärenten Konservatismus und Inkrementalismus hin, der mit dem Bezug auf Autoritäten einhergeht. Hinzu kommen die Risiken willkürlicher Entscheidungen entlang sachlich unangemessener oder intransparenter Kriterien, der Überlastung der zentralen MacherInnen, deren Zeit und Aufmerksamkeit vielfach zur Engstelle werden (vgl. Paul 1999), und auf Seiten der ZuarbeiterInnen die Risiken

10 Die Darstellung folgt der Übersicht bei Bode 1997: 54ff.

11 Das ist nicht weit von anderen Thesen der Aufmerksamkeitsökonomie entfernt, wie sie für das Internet etwa von Goldhaber (1997), Florian Rötzer (1998) oder auch Peter Glotz (1999) vertreten werden. Zum steigenden Gewicht der außerordentlichen Leistung und der Reputation gegenüber „normaler" Aufgabenerfüllung auch in der Erwerbsarbeit s. Voswinkel 1999 und die Beiträge in Holtgrewe/Voswinkel/Wagner (Hrsg.) 2000.

der Überidentifikation mit dem Projekt, die sie für Missachtungserfahrungen anfälliger macht. Gleichzeitig kann die Personalisierung, die sich mit der Reputationsökonomie verbindet, den Blick für eben solche Krisen und Restriktionen struktureller Art verstellen. Die veränderten Anforderungen an die Linux-Entwicklung für einen kommerziellen Markt (s. Kap. 4.2) etwa, die die Entwicklung merklich verlangsamt haben, werden durchaus entlang der Frage „Ist Linus Torvalds überfordert?“ diskutiert, die diese Probleme *ad personam* zurechnet (vgl. die Materialien in Bezroukov 2000).

Die Reputationsökonomie eröffnet den Raum für eine weitere zweckrationale Erklärung der Teilnahme an OS-Projekten: Reputation kann über die *community* hinaus in die Sphäre der Erwerbsarbeit übertragen werden, so dass erfolgreiche Open-Source-Beiträge für Software-EntwicklerInnen auf manchen Gebieten eine durchaus rationale Investition in die eigene Karriere sein können. Gratisarbeit für die Reputation und den Aufbau von Beziehungen wäre dann vergleichbar mit den Berufseinstiegs- und Umstiegsstrategien in anderen „kreativen“ Branchen wie Journalismus, Architektur, Kunst usw. (Haake/Schmid 1999, Nardi/Whittaker/Schwarz 2000), wobei die Arbeitsmarktsituation für Computerfachleute dies zu einer stärker optionalen Strategie macht.

Eine stärker gebrauchswertorientierte Deutung der Kollektivguterstellung hat der indische Ökonom Rishab Aiyer Ghosh (1998) mit seinem Modell einer „Kochtopfökonomie“ vorgelegt. Damit ist eine Infrastruktur sozialen Austauschs gemeint, in der man für seinen eigenen begrenzten Beitrag den Zugang zu einer Vielfalt von Beiträgen bekommt. In der Nutzung eben dieser Vielfalt, den Chancen der Betätigung, des Lernens und des Vergnügens liegt der spezifische Nutzen der Teilnahme. Weil die Grenzkosten der Verbreitung der Güter im Falle von *Open Source* vernachlässigbar sind, sind Trittbrettfahrer kein Problem – aber sie verpassen den Hauptnutzen und Gewinn des Projekts.

Nicht zu vernachlässigen – und in der Theorie kollektiven Handelns eher ausgeblendet – sind die intrinsischen Motive. Programmieren ist nach Aussagen der Entwickler eine Tätigkeit, die hinreichenden Spaß macht, dass sie nicht unbedingt entlohnt oder durch Anreize motiviert werden muss. Dabei schließen Intrinsik und Reputationsökonomie aneinander an. Eine Grundlage für das Streben nach Anerkennung ist hier wie auch anderswo eine ganz kindlich-spielerische Begeisterung: „Look at what I‘ve done – isn’t this neat?“ (der Linux-Gründer Linus Torvalds im Interview mit First Monday 1998).

Aus dieser Sicht des rein technischen Produzentenstolzes und der Reputation wird die Frage der wirtschaftlichen Verwertung durch andere problemlos:

„ Und als NCR vor vielen Jahren als erste Firma gesagt hat, ‚Wir benutzen euren Server [XFree86] in unserem kommerziellen Unixsystem. Habt ihr damit ein Problem?‘ da war unsere Antwort: ‚Oh, super, klasse! Dürfen wir das anderen erzählen?‘“ (Dirk Hohndel, Mitglied des Kernteams von XFree86, in www.mikro.org/Events/OS/ref-texte/disk_lizenzen)

Wenn man sich auf technische Exzellenz bezieht, gönnt man anderen gern den wirtschaftlichen Gewinn – und das mag dann erleichtert werden, wenn man sich auf einem vergleichsweise entspannten Arbeitsmarkt befindet. Im Gegensatz zu dieser herzerfrischenden Sicht regeln sich zurzeit die Beziehungen an den Schnittstellen zur Welt kommerzieller Soft- und Hardwareentwicklung nicht (mehr) ganz so unkompliziert. Es entstehen komplexe Vernetzungen und Verhandlungen zwischen Industrie und *Open Source* (s. Absatz 4.2).

3.2 Pioniere zwischen Autarkie und gegenseitiger Hilfe

Die Programme, Selbstdeutungen und Überlieferungen der Open-Source-*communities* kultivieren ein spezifisches Selbstbild und eine entsprechende Vorstellung von Gemeinschaften jenseits und entgegen der „normalen“ Art der Technikentwicklung und Erwerbsarbeit: Im Grunde sehen sich die EntwicklerInnen als potenziell autarke ProduzentInnen, die über alle Fähigkeiten verfügen (wollen), auch ihre Produktionsmittel selbst herzustellen. Wer etwa auf Kongressen das Fehlen bestimmter Programme oder Features moniert, wird freundlich aufgefordert, diese doch selbst zu entwickeln und dazu ggf. eben Programmieren zu lernen (www.mikro.org/Events/OS/ref-texte/disk_lizenzen). Weil aber jedeR mehr gute Werkzeuge

gebrauchen als herstellen kann und die Distribution der entwickelten Werkzeuge geringen Aufwand verursacht, werden diese der Gemeinschaft zur Verfügung gestellt. Auch in der Nutzung gelten Normen der gegenseitigen Hilfe: Es werden Ratschläge gegeben, Fragen beantwortet, Verbesserungen vorgenommen. Es wird also einerseits wechselseitig Kompetenz unterstellt und gefordert, andererseits respektiert, dass nicht alle alles können müssen. Personen werden als AutorInnen anerkannt, aber die unvermeidlichen Programmfehler werden nicht ihnen zugerechnet, sondern als sachlich unvermeidlich bearbeitet. Das ähnelt – zumindest in den etwas idealisierenden Selbstbeschreibungen (Raymond 1999a) – einer Siedlergemeinschaft und greift auf entsprechende us-amerikanische kulturelle Bestände zurück.

Eine Gemeinschaft, die sich so versteht, grenzt ihre potenzielle Offenheit auch wieder ein – entlang der Kriterien knapper Zeit und Aufmerksamkeit. Ein gewisses Maß an Kompetenz sollten die AnfängerInnen und NutzerInnen nicht unterschreiten. Während der Support im Sinne gegenseitiger Hilfe per Usenet und Newsgroups durchaus gepflegt wird, kann nicht von jedeR EntwicklerIn erwartet werden, dass sie jede „dumme Frage“ geduldigst und erschöpfend beantwortet. Hier ist eine Arbeitsteilung im Interesse aller Beteiligten, die etwa den Anfänger-Support zur kommerziellen Dienstleistung werden lässt.

Auch zu Konflikten und Krisen in den Projekten kommt es dann, wenn zu viele „Ahnungslose“ mitreden wollen, zu viel diskutiert und zu wenig entwickelt wird. Es scheint, dass die EntwicklerInnen darauf mit Aufmerksamkeitsbegrenzung und selektivem Ignorieren der entsprechenden Mailinglisten reagieren – was aber wieder Frustrationen und Projektkrisen erzeugt (Lehmann 1999).

Dass es vielen Projekten gelingt, über lange Zeit stabil funktionierende Software zu entwickeln, erfordert also sozial beachtliche Balancen (die natürlich auch scheitern können): zwischen Problemen und Lösungen, zwischen Konzentration auf „den Code“ und Reflexion der Entwicklung des gesamten Projekts, zwischen technischen Standards und sozialen Normen und Verkehrsformen. Wohlgemerkt geschieht all dies mittels elektronischer, also medial ausgedünnter Kommunikation zwischen räumlich getrennten Individuen.[12]

3.3 Professionelle BeobachterInnen: *Open Source* als Irrweg oder Lernchance

Beobachter aus der professionellen Softwareentwicklung werden durch das Interesse und Engagement von EntwicklerInnen und insbesondere Studierenden in Open-Source-Projekten eher befremdet. Manch einer vermutet bei den EntwicklerInnen grundlegend irrationale, „romantische“ oder adoleszent-rebellische Motive (Endres 2000; Glass 2000). Unter dem irreführenden Titel „The Sociology of Open Source“ bemerkt Robert L. Glass (2000), dass die Arbeitsweisen der Open-Source-Projekte im Gegensatz zu all seinen Kenntnissen der Motive und Neigungen von Software-EntwicklerInnen stünden. Diese interessierten sich im allgemeinen gerade nicht dafür, Code zu studieren und zu pflegen und würden eigentlich gern gutes Geld verdienen (S. 105). Endres (2000) fordert in der Zeitschrift Informatik-Spektrum, dem Organ der deutschen „Gesellschaft für Informatik“, die Softwareentwicklung dem Markt bzw. der Kompetenz „normaler“ Softwareunternehmen vorzubehalten. Er plädiert für saubere Trennungen zwischen Entwicklung und Nutzung. *Open Source* als Entwicklungsmodell erscheint dann als ökonomisch und professionell potenziell gefährlicher Irrweg oder als „Kult“ (Glass 2000). Dabei beziehen auch diese Autoren sich eher auf die skizzierten, durchaus idealisierenden Selbstdarstellungen und Programme der Open-Source-*community*, als auf die Erfahrungen existierender Projekte. Die Tatsache funktionierender und erfolgreicher Entwicklungen können sie auf diesem Wege nicht erklären.

Demgegenüber haben neuerdings Moon und Sproull (2000) die Bedingungen erfolgreicher Open-Source-Projekte am Beispiel Linux zu identifizieren versucht, um sie auf die Organisation verteilter (Entwicklungs-)Arbeit in Unternehmen zu übertragen. Solche Möglichkeiten

12 Ich vermute allerdings, dass persönliche Kontakte, räumliche Schwerpunkte, Treffen und Ereignisse *in real life* diese Projekte nicht unbeachtlich strukturieren. Die Rolle von Messen, Konferenzen und informellen Treffen dürfte nicht zu unterschätzen sein.

sehen sie in der modularen, und damit transparenten und zugänglichen Beschaffenheit der Produkte und Arbeitsaufgaben, in Anreizsystemen, die wie die Reputationsökonomie das Teilen von Informationen fördern, und in einer expliziten Pflege und Reflexion „elektronischer Rollen“, Erwartungen und Normen als Bestandteil elektronischer Kooperationszusammenhänge.

Berichte aus einzelnen Projekten (s. www.mikro.org/Events/OS/ref-texte, Kuwabara 2000) legen dabei nahe, dass die Vielfalt an Organisations- und Koordinationsformen größer ist, als das im Ensemble aus charismatischen GründerInnen, erfolgreicher Etablierung von Reputationsmärkten und der sozialen Integrationskraft der Hacker-Kultur (Raymond 1998; Moon/Sproull 2000) zum Ausdruck kommt. Zur Konsens stiftenden Intrinsik des Programmierens kommen vermutlich die Eigenlogik und Verbindlichkeit von Projekten, aber auch gemischte und komplexe Motive und Interessen: Professionalität *und* Altruismus, berufliche Anschlussmöglichkeiten *und* die Gelegenheit, etwas anderes, zur Erwerbsarbeit Komplementäres zu tun, technische Exzellenz *und* Gegenkultur. Im folgenden gehe ich daher den Wechselwirkungen und Durchlässigkeiten zwischen technischer und sozialer Innovation nach.

4. Soziale Transformation und technische Innovation: AktivistInnen und IngenieurInnen

Wenden wir uns nun zwei Beispielen zu, die entlang der Frage der Transformation oder Reproduktion gesellschaftlich etablierter Erfolgskriterien einerseits den Aspekt sozialer Bewegung, andererseits den „normaler“ technischer Innovation verdeutlichen. Hier liegt der Unterschied zwischen *Open Source* im engeren Sinne und freier Software. Den Unterschied kann man als einen zwischen „IngenieurInnen“ überlegener Technik und „AktivistInnen“ sozialer Transformation zuspitzen.[13]

Die „IngenieurInnen“ beziehen sich auf die technisch-organisatorische Überlegenheit des Open-Source-Modells – und gehen davon aus, dass sich technische Exzellenz am Erfolg erweist. Man sieht Erfolg in Begriffen von Markt- bzw. Nutzungsanteilen in direkter Konkurrenz zur proprietären Software, oder in der Entscheidung kommerzieller Soft- und Hardwarehersteller, Quellcodes oder Hardwarespezifikationen offenzulegen. „Philosophische“ Ansprüche und Programme werden dabei für eher reputationsschädlich erklärt, in den Worten Eric Raymonds (des Haupt-Programmatikers): „Shut up and show them the code!“ (1999b).

Die AktivistInnen um die Free Software Foundation um Richard M. Stallman verstehen „Freiheit“ als weit gehenden sozialen und philosophischen Anspruch, der vom Gebrauchswert und der intelligenten Nutzung der Produkte her gedacht wird. Diese sind Gegenstände des sozialen Austauschs und der Kommunikation. Institutionen, die die kreative Aneignung und Weiterentwicklung behindern, werden deswegen kritisiert. Von der Norm der Kreativität aus gehend also geht es um die Umgestaltung der Institutionen geistigen Eigentums.[14]

4.1 (Selbst-)Bindung an Freiheit: Die General Public License

Die soziale Innovation, an der die transformativen Ambitionen der AktivistInnen am deutlichsten werden, ist das so genannte Copyleft. Das ist eine juristische Konstruktion einer Lizenz, der General Public License (= GPL), unter der bestimmte Open-Source-Software – übrigens auch Linux – vertrieben wird. Sie sieht nicht nur vor, dass die Produkte frei und kostenlos verbreitet und weiterentwickelt werden dürfen. Sie erlaubt dies unter der Bedingung, dass

13 Beide Gruppen haben sich in Stiftungen organisiert, der Open Source Foundation und der Free Software Foundation, die sich neben eigenen Projekten und Beiträgen die öffentliche Repräsentation und Interessenvertretung der Projekte zur Aufgabe gemacht haben.

14 Auch die normative Freiheitsperspektive ist jedoch nicht antikommerziell. Es ist vollkommen legitim, mit Dienstleistungen um die kollektiv erstellten Produkte herum Geld zu verdienen: Kommerzielle Dienstleistungen und öffentliche Güter sollen einander – in durchaus produktiver Konkurrenz – komplementieren.

diese Entwicklungen und Verwendungen wieder unter dieselbe Lizenz gestellt werden. Es handelt sich also um eine trickreiche Konstruktion, die sich der Mittel des Copyright bedient, um es aufzuheben. Das Copyright räumt ja der RechteinhaberIn ein, über die Art und Weise der Verbreitung und Verwendung ihrer kreativen Erzeugnisse zu bestimmen. Das Copyleft („All rights reversed“) bindet die weiterhin produktive und kreative Nachwelt an die Bedingungen offener Nutzung – eine Selbstbindung an Freiheit, der Versuch, sich und die Nachwelt auf Vielfalt festzulegen.

So clever diese Konstruktion juristisch und sozial ist, ist sie faktisch nicht zum Standard der Open-Source-*community* geworden. Viele wichtige Projekte (wie BSD, Apache, Perl usw.) haben ihre eigenen Lizenzen kreiert, die gerade diese Fortpflanzungsfähigkeit oder den „ansteckenden“ Charakter der GPL beschränken.[15] Gründe dafür sind nicht einfach die Verwertungsinteressen der Projekte. Vielmehr werden die Anforderungen der GPL gerade in Kooperationen mit Wirtschaftsunternehmen zum konkreten Problem, und auf diese will oder kann man nicht verzichten. Solche Kooperationen sind notwendig, wenn man beispielsweise Schnittstellen zu proprietären Programmen vorsieht oder Hardwaretreiber schreibt. Dazu braucht man Informationen der Hersteller, und diese sind zwar durchaus daran interessiert, diese zur Verfügung zu stellen, aber nicht daran, ihre Eigentumsinteressen aufzugeben und sich an die Ökonomie öffentlicher Güter zu binden.

Deswegen stehen die Normen der GPL – wenn ich die Diskussionen der *community* richtig interpretiere, vgl. etwa www.mikro.org/Events/OS/ref-texte/disk_lizenzen und die Newsgroup gnu.misc.discuss – in Konflikt mit einer anderen, sozial und technisch vielfach institutionalisierten Norm der Internet-Communities allgemein: der Norm der Interkonnektivität (vgl. Dyson 1997; Helmers u. a. 1998; Werle/Leib 1999, Werle 2000), also der Sicherstellung der Möglichkeit vielfältiger Verbindungen zwischen technisch und sozial unterschiedlichen Netzen, Geräten und Programmen. Konsens besteht über offene Standards, aber der Konsens erstreckt sich offensichtlich dann nicht auf die Selbstbindung an öffentliche Güter, wenn diese mögliche Verbindungen ausschließt. Auch eine Bindung an Freiheit wird dann als Einschränkung abgelehnt.

So betrachtet ist die begrenzte Verbreitung der GPL nicht als eine vollzogene Domestizierung und Anpassung von *Open Source* und freier Software zu betrachten, in der die AktivistInnen gegenüber den IngenieurInnen erwartbar den Kürzeren gezogen haben. Zwar haben sich die Projekte zwischen dem Bezug auf soziale Transformation und dem auf realexistierende Unternehmen und Professionen auch personell und institutionell ausdifferenziert. Beide Seiten bleiben jedoch in Verbindung. Mit der sozialen Innovation des Copyleft haben die AktivistInnen die Maßstäbe verschoben und einen normativen (Maximal-)Standard gesetzt. Die GPL macht die Erhaltung der Bedingungen kreativer Betätigung zum Thema, mobilisiert die soziale Fantasie und schließt sie an die technischen Möglichkeiten an. Damit zeigt sie die sozialen Alternativen zu einer „New Economy“, in der Austauschregeln sich an marktlicher Fairness orientieren und die Freiheit lediglich in der Wahlmöglichkeit der Konsumentin besteht.

4.2 Linux: Riskanter Erfolg

Den Kontrastfall eines Erfolges im Sinne der Ingenieurs-Orientierung stellt Linux dar.[16] Bei Linux handelt es sich zunächst einmal um den „Kern“ eines freien, unixkompatiblen Betriebssystems. Ein solches komplettes System zu entwickeln, war von Anfang an auch Ziel der AktivistInnen der Free-Software-Bewegung. Mit der Kernentwicklung kam ihnen jedoch Linus Torvalds, damals Student an der Universität Helsinki, 1991 zuvor. Er machte sich daran, einen Kern für seinen 386er PC zu entwickeln und warb um MitstreiterInnen.

15 Perens (1999) gibt eine Übersicht über die Konkurrenzen und Kompromisse zwischen verschiedenen Lizenzmodellen, an der die Wechselwirkungen zwischen Maximalstandard und anderen (Verwertungs)-Interessen deutlich werden.

16 Die Darstellung folgt Bezroukov 2000, vgl. aber auch Raymond 1998; Kuwabara 2000; Grassmuck 2000; Moon/Sproull 2000.

``Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on an OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.“ (Usenet-Newsgroup comp.os.minix, zit. nach Bezroukov 2000)

Es fand sich tatsächlich Unterstützung aus diversen EntwicklerInnencommunities und das ganze entwickelte und organisierte sich um Torvalds als zentralen Entscheider und Koordinator.

Beim erfolgreichen Start des Projekts trafen verschiedene, durchaus kontingente Faktoren zusammen. Es gab die Personen mit den entsprechenden Fähigkeiten und Bereitschaften, die teils von kriselnden Konkurrenzprojekten absprangen. Mit Minix, einem kleinen, proprietären Betriebssystem, das für den Einsatz in der Informatikerausbildung geschrieben worden war und sich einer großen, lebhaften Entwicklergemeinde erfreute, lag ein Modell vor, an dem man sich abarbeiten konnte. Hinreichend leistungsfähige PCs (386er) und brauchbare Bücher über Unix waren vorhanden.

Den Normen der Hackerkultur entsprechend stellte Torvalds Linux – wohlgemerkt – unter die GPL. Auf dieser Grundlage wurde es zum Paradebeispiel dezentraler, kooperativer Entwicklung, unter einer „wohlwollenden Diktatur“ von Torvalds und alsbald einem Kernteam, das Unterprojekte in Gang hielt, mit schnellen und häufigen Releases, die ebenso schnell getestet und verbessert wurden.

Ab ca. 1993/1994 wurde dabei deutlich, wie der Erfolg des Projekts es in andere Netzwerke, unter andere Gelegenheiten und Restriktionen brachte. Unterstützung im Verbund mit kommerziellen Interessen kam von Hardwareherstellern, die ebenso wie die Linux-Entwickler an Portierungen auf ihre Hardware und an Treibern interessiert waren. Vielfach betrieben EntwicklerInnen in den Firmen diese Öffnung. Sodann wurden – vielfach von Linux-EntwicklerInnen - die ersten Distributionsfirmen gegründet, die Dienstleistungen um die Software herum anbieten. Die verschiedenen Bestandteile von Linux auszuwählen, Kern, Utilities, Installationsroutinen usw. auf einem Rechner „zusammenzubauen“ und zum Laufen zu bringen, ist nämlich nicht ganz einfach und erfordert einiges an Programmierkenntnissen und Zeit. Aus diesem Grund gibt es Distributionen, die gewissermaßen vorbereitete Pakete schnüren und seit 1993 zunehmend von kommerziellen Firmen vertrieben werden.[17] Diese Firmen stellten Linux-EntwicklerInnen ein, die weiter an freien Projekten arbeiten.[18] Sie investieren also trotz oder wegen der GPL auch selbst in die Weiterentwicklung von Linux mit dem Interesse, das Betriebssystem und damit auch ihre Distributionen attraktiver zu machen und auch unter der Perspektive, einen fairen sozialen Tausch mit *der non-profit-community* einzuhalten.

Dazu entstanden Zeitschriften und Verlage, so dass sich Verbreitung und Reputation des Betriebssystems, das nunmehr unter dem Markennamen Linux lief (den Torvalds als Warenzeichen hat schützen lassen) gegenseitig steigerten. Dabei war einerseits die freie und kooperative Geschichte von Linux sowohl technisch als auch kulturell Teil der Attraktivität, andererseits verschob sich die öffentliche Aufmerksamkeit auf die Gründerfigur, die zum „Erfinder“ mutierte.

Mit dem Erfolg von Linux nehmen absehbar die Reibungen zwischen Profit und Nonprofit-Sphären zu. Der Einfluss kommerzieller Firmen steigt, aber auch der Anteil „laienhafter“ NutzerInnen, privater wie Unternehmen, wird größer. Diese sind bei Problemen nicht mehr in der Lage, ihre eigenen Verbesserungen vorzunehmen und weiterzugeben oder sich fehlende Treiber selbst zu schreiben. Ihre Erwartungen an Support und Features entsprechen eher denen an kommerzielle Software. Das ist zwar das Betätigungsfeld der Distributions- und Beratungsfirmen, aber auch die ehrenamtlichen EntwicklerInnen erfahren zunehmend die *constraints* der Konkurrenz mit kommerzieller Software. Der Zeitdruck nimmt zu, die Komplexität auch – eine klassische Zeitfalle der Innovation (Rammert 1997) –, und gerade der Erfolg auf dem Markt führt zu Frustrationen der „Ehrenamtlichen“, wenn ihre Beiträge nicht aufgegriffen werden und NutzerInnen sich zunehmend als Kunden statt als Kooperationspartner

17 Es gibt auch eine nichtkommerzielle Distribution, nämlich debian/GNU.

18 Linus Torvalds selbst ist seit 1997 beim – notorisch geheimniskrämerischen – Chiphersteller Transmeta beschäftigt, aber ist weiterhin der zentrale Linux-Entscheider.

verhalten. So wird durchaus diskutiert, ob die Koordinationsstruktur der „wohlwollenden Diktatur“ nicht die zentralen Personen überfordert und das Projekt gefährdet (Bezroukov 2000).

Ein möglicherweise gewichtiger Kommerzialisierungs- oder Normalisierungsschritt hat im September 2000 stattgefunden: Der Nachrichten-Dienst Linuxgram berichtet am 1. Sptember, dass die großen Linux-Distributionsfirmen (RedHat, VA Linux, TurboLinux und SuSE) gemeinsam mit Hardwareherstellern wie Intel, IBM, NEC und HP ein neues Open-Source-Konsortium (Open Source Development Lab) mit zweistelligen Dollar-Millionen gründen, das ein Entwicklungslabor betreiben wird. Das Ziel wird sein, Linux zu einem vollwertigen Betriebssystem für kommerzielle Anwendungen auszubauen und wohl auch die Auswirkungen der GPL in Grenzen zu halten (www.linux.com/linuxgram/newsitem.phtml?sid=1098aid=10562).

Auch hier jedoch bleiben die Reaktionen der Community abzuwarten. Bislang hat sich gezeigt, dass die Szene durchaus und zu Recht erwartet, dass die Firmen, die ihr Geld um *Open Source* herum verdienen, Gegenleistungen erbringen. Sie bleiben unter oftmals strenger Beobachtung und sozialer Kontrolle. Wenn dies so bleibt, also die AktivistInnen Linux nicht als vollendeten Kommerzialisierungsfall abschreiben, mag das Entwicklungslabor den Impuls geben, auch die nichtkommerzielle Seite z. B. in Stiftungsform zu institutionalisieren (so Vorschläge von Bezroukov 2000 und Browne 2000).

5. Fazit: Nachhaltige Kreativität und Pflege der Vielfalt

Wir haben also gesehen, dass in einem für den Internetbetrieb insgesamt nicht unbedeutenden Bereich technischer Innovation die Orientierungen und kreativen Erzeugnisse sozialer Bewegungen weiterhin eine Rolle spielen. Die AktivistInnen versuchen, gerade die Normen der Kreativität in Gestalt des freien, sozialen Tausches von Gebrauchswerten zu bestimmen, und gemäß diesen Kriterien werden die Institutionen geistigen Eigentums umgebaut und bewertet. Die „IngenieurInnen“ schließen die Kriterien technischer Exzellenz an die Erfolgskriterien des Marktes an – geraten dabei aber auch unter die Zwänge des Marktes.

Dabei scheint es, dass AktivistInnen und IngenieurInnen sich bislang nicht in Flügelkämpfen verzetteln. In der Tat profitieren beide Seiten von den unterschiedlichen Erfolgen der jeweils anderen: Linux und die anderen weit verbreiteten Betriebssysteme und Programme haben den Free-Software-AktivistInnen den Beweis geliefert, dass wettbewerbsfähige Softwareentwicklung auf nichtkommerzielle Weise möglich ist und dass technische Innovationen auch als öffentliche Güter entstehen. Dass Linux unter der GPL vertrieben wird, hat gerade als Selbstbindung der *community* die Attraktivität für EntwicklerInnen und NutzerInnen erhöht und auch für die Distributionsfirmen die Bedingungen und Risiken der Vermarktung kalkulierbar gemacht: Wenn sie es nicht privatisieren können, so können ihre Wettbewerber das auch nicht.

Die Etablierung der Kreativität als Norm in den sozial bewegten Netzwerken der Softwareentwicklung führt also nicht bruchlos zu einer Orientierung am Markterfolg als Bewertungskriterium kreativen Handelns. Eine Logik der Kommerzialisierung und „Normalisierung“ ist am Beispiel Linux aufzufinden, doch bleibt diese eingebettet in soziale Tauschbeziehungen und normative Orientierungen der Non-Profit-Seite. Auf dieser Seite werden die Normen der Kreativität reflexiv: Sie werden auf die Institutionen des Innovationssystems selbst angewandt, und dieses wird – in gewissermaßen „nachhaltiger“ Weise – auf seine Möglichkeiten der Erhaltung und des Ausbaus der Produktions- und Kooperationsbedingungen kreativen Handelns hin evaluiert. Die wirtschaftlichen und moralischen „Ökonomien“ der *Open Source* und freien Software also bleiben miteinander verzahnt in spannungsvollen und spannenden Verhältnissen, in denen die Grenzen zwischen Märkten und Projekten, Kommerz und Non-Profit erst gezogen, verschoben und verhandelt werden.

Wenn also in der Tat die Koordinations- und Kooperationsmechanismen der „Wissensgesellschaft“ im Entstehen sind, stellt sich die „New Economy“ als nur ein (bekanntlich in den

letzten Monaten etwas gerupfter) Teil davon dar. Dann macht es Sinn, den Kurzschluss kreativen Handelns mit dem Markt nicht antizipierend festzuschreiben, sondern sich mit einer Kombination von Perspektiven auf die Vielfalt der Deutungen und normativen Grundlagen einzulassen, die hier immer noch und weiterhin im Spiel sind. Hier sind in der Tat „Formen sozialer Odnung im Prozeß der Selbsterzeugung“ zu beobachten (Joas 1996: 291). Dieser Beitrag war ein Versuch dazu.

Literatur

Baecker, Dirk (1997), Was ist Kreativität? In: taz vom 2.8.1997: 6.

Barbrook, Richard (1998), The Hi-Tech Gift Economy, in: First Monday 3, http://www.firstmonday.org/issues/issue3-12/barbrook/index.html.

Barker, Colin/Kennedy, Paul (Hrsg.) (1996), To Make Another World. Aldershot u. a.

Bender, Gerd (1999), Wissensarbeit und Innovationsnetzwerke. Forschungsfragen und Konzeptualisierungsvorschläge. Beitrag zur Tagung „Wissensarbeit“ der DGS-Sektion Industrie- und Betriebssoziologie am 5. und 6. November 1999 in Duisburg, Dortmund (unv. Ms.).

Bezroukov, Nikolai (1999a), Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism), in: First Monday 4, http://www.firstmonday.org/issues/issue4_10/bezroukov/index.html.

Bezroukov, Nikolai (1999b), A Second Look at the Cathedral and the Bazaar, in: First Monday 4, http://www.firstmonday.org/issues/issue4_12/bezroukov/index.html.

Bezroukov, Nikolai (2000), Portraits of Open Source Pioneers Kap. 4.1: Linus and Linux: Linus Torvald’s Short Unauthorized Biography, www.softpanorama.org/People/Torvalds/Linus_Torvalds_biography.shtml Download vom 5.9.00.

Bode, Ingo (1997), Die Organisation der Solidarität: Normative Interessenorganisationen der französischen Linken als Auslaufmodell mit Zukunft. Opladen.

Browne, Christopher B. (1998), Linux and Decentralized Development, in First Monday 3, http://www.firstmonday.org/issues/issue3_3/browne/index.html.

DiBona, Chris/Ockman, Sam/Stone, Mark, (Hrsg.) (1999), Open Sources: Voices from the Open Source Revolution. Cambridge, Mass. u. a., http://www.oreilly.com/catalog/opensources/book/toc.html.

Dyson, Esther (1997), Release 2.0. Die Internet-Gesellschaft. Spielregeln für unsere digitale Zukunft. München.

Elster, Jon (1987), Subversion der Rationalität. Frankfurt/Main, New York.

Endres, Albert (2000), „Open Source“ und die Zukunft der Software, in: Informatik Spektrum 23: 316–321.

FifF-Kommunikation 3/99 Open Source(s) – Offene Quellen..

First Monday (1998), FM Interview with Linus Torvalds, in: First Monday 3, http://www.firstmonday.org/issues/issue3_3/torvalds/index.html.

Ghosh, Rishab Aiyer (1998), Cooking Pot Markets: An Economic Model for the Trade in Free Goods and Services on the Internet, in: First Monday 3, http://www.firstmonday.org/issues/issue3_3/ghosh/index.html.

Ghosh, Rishab Aiyer/Prakash, Vipul Ved (2000), The Orbiten Free Software Survey 1st edition, May 2000, http://orbiten.org/ofss/01.htm.

Glass, Robert L. (2000), The Sociology of Open Source: Of Cults and Cultures, in: IEEE Software 17(3): 104f..

Glotz, Peter (1999), Die beschleunigte Gesellschaft. Kulturkämpfe im digitalen Kapitalismus. München.

Goldhaber, Michael M. (1997), The Attention Economy and the Net, in: First Monday 2, http://www.firstmonday.org/issues/issue2-4/goldhaber/.

Grassmuck, Volker (2000), Open Source – Betriebssystem für eine freiheitliche Gesellschaft. Vortrag auf der Tagung „Freie Software – Ein Modell für die Bürgergesellschaft“, Evangelische Akademie Tutzing 31. Mai – 1. Juni 2000. http://www.waste.informatik.hu-berlin.de/grassmuck/texts/oss-tutzing-5-00.html

Groys, Boris (1997), Technik im Archiv. Die dämonische Logik technischer Innovation, in: Bechmann, Gotthard/Rammert, Werner (Hrsg.): Jahrbuch Technik und Gesellschaft 9: Innovation: Prozesse, Produkte, Politik. Frankfurt/Main, New York: 15-32.

Hack, Lothar (1988), Vor Vollendung der Tatsachen. Die Rolle von Wissenschaft und Technologie in der dritten Phase der Industriellen Revolution. Frankfurt/Main

Helmers, Sabine/Hoffmann, Jeanette/Hofmann, Ute (1998), Internet ... the Final Frontier: eine Ethnographie. Schlußbericht des Projekts „Interaktionsraum Internet – Netzkultur und Netzwerkorganisation“ WZB- discussion paper FS II 98-112. http://duplox.wz-berlin.de/endbericht/index.htm.

Holtgrewe, Ursula (2000), Recognition, Intersubjectivity and Service Work: Beyond Subjectivity and Control. Contribution to the 18th Annual International Labour Process Conference, 25 – 27 April 2000, University of Strathclyde, Glasgow, http://soziologie.uni-duisburg.de/PERSONEN/holtgrewe/uh-lpc2k.pdf.

Holtgrewe, Ursula/Voswinkel, Stephan/Wagner, Gabriele (2000), Für eine Anerkennungssoziologie der Arbeit. Einleitende Überlegungen, in: Holtgrewe, Ursula/ Voswinkel, Stephan/Wagner, Gabriele (Hrsg.): Anerkennung und Arbeit. Konstanz: 9-26.

Holtgrewe, Ursula/ Voswinkel, Stephan/Wagner, Gabriele (Hrsg.) (2000): Anerkennung und Arbeit. Konstanz.

Honneth, Axel (1994), Kampf um Anerkennung. Zur moralischen Grammatik sozialer Konflikte. Frankfurt/Main.

Joas, Hans (1996), Die Kreativität des Handelns. Frankfurt/Main.

Krafft, Alexander/Ortmann, Günther (Hg.) (1988), Computer und Psyche. Angstlust am Computer. Frankfurt/Main.

Kuwabara, Ko (2000), Linux: A Bazaar at the Edge of Chaos, in: First Monday 5, http://www.firstmonday.org/issues/issue5_3/kuwabara/index.html.

Lehmann, Marc (1999), The GIMP. Beitrag zum Kongress „Wizards of OS: Offene Quellen und freie Software“ in Berlin 16./17. Juni 1999, www.mikro.org/Events/OS/ref-texte/lehmann.html.

Levy, Steven (1984/1994), Hackers. Heroes of the Computer Revolution. New York.

Marwell, Gerald/Oliver, Pamela (1993), The Critical Mass in Collective Action: A Micro-Social Theory. Cambridge.

Meretz, Stefan (2000), LINUX & CO. Freie Software – Ideen für eine andere Gesellschaft. Version 1.01, letzte Änderung: 03.07.2000, http://www.kritische-informatik.de/fsrevol.htm.

Moldaschl, Manfred/Sauer, Dieter (2000), Internalisierung des Marktes – Zur neuen Dialektik von Kooperation und Herrschaft, in: Minssen, Heiner (Hrsg.): Begrenzte Entgrenzungen. Wandlungen von Organisation und Arbeit. Berlin: 205-224.

Moon, Jay Yun/Sproull, Lee (2000), Essence of Distributed Work: The Case of the Linux Kernel, in: First Monday 5, http://www.firstmonday.org/issues/issue5_11/moon/index.html, Download 12.11.2000.

Nardi, Bonnie A./Whittaker, Steve/Schwarz, Heinrich (2000), It’s Not What You Know, It’s Who You Know: Work in the Information Age, in: First Monday 5, http://www.firstmonday.org/issues/issue5_5/nardi/index.html, Download 23.8.2000.

Olson, Mancur (1968), Die Logik des kollektiven Handelns : Kollektivgüter und die Theorie der Gruppen. Tübingen.

Paul, Gerd (1999), An der Wissensarbeiterfront? Das Beispiel der industriellen Softwareproduktion, in: Konrad, Wilfried/Schumm, Wilhelm (Hrsg.): Wissen und Arbeit. Neue Konturen von Wissensarbeit. Münster: 77-91.

Perens, Bruce (1999), The Open Source Definition, in: DiBona, Chris/Ockman, Sam/Stone, Mark, (Hrsg.), Open Sources: Voices from the Open Source Revolution, Cambridge, Mass. u. a. (O’Reilly), http://www.oreilly.com/catalog/opensources/book/toc.html.

Rammert, Werner (1997), Innovation im Netz. Neue Zeiten für technische Innovationen: heterogen verteilt und interaktiv vernetzt, in: Soziale Welt 48: 397-416.

Raymond, Eric (1998a), The Cathedral and the Bazaar, in: First Monday 3, http://www.firstmonday.org/issues/issue3_3/raymond/index.html.

Raymond, Eric (1999a), Homesteading the Noosphere. Version August 1999, http://www.tuxedo.org/~esr/writings/homesteading/homesteading.txt.

Raymond, Eric (1999b), Shut Up and Show Them the Code. Version 9.7.1999, http://www.tuxedo.org/~esr/writings/shut-up-and-show-them.html.

Rötzer, Florian (1998), Hacker und Aufmerksamkeitsökonomie. Zu Eric Raymonds Kultur des Schenkens, http://www.heise.de/tp/de...al/auf/2532/1.html.

Schmid, Johannes K. (1997), Der Personal Computer (1974 – 1985). Architektonische Innovation und vertikale Desintegration, in: Weyer u. a.: Technik, die Gesellschaft schafft. Soziale Innovation als Ort der Technikgenese. Berlin: 147–226.

Schwendter, Rolf (1973/1981), Theorie der Subkultur. Frankfurt.

Strübing, Jörg (1993), Subjektive Leistungen im Arbeitsprozeß. Eine empirische Untersuchung von Arbeitsstilen in der Programmierarbeit. Wiesbaden.

Voss, G. Günter/Pongratz, Hans J. (1998), Der Arbeitskraftunternehmer. Eine neue Grundform der Ware Arbeitskraft? In: Kölner Zeitschrift für Soziologie und Sozialpsychologie 50: 131-158

Voswinkel, Stephan (1999), Anerkennung und Reputation. Die Dramaturgie industrieller Beziehungen. Duisburg (unv. Habilitationsschrift).

Wenger, Etienne (1998), Communities of Practice. Cambridge.

Werle, Raymund (2000), Das „Gute“ im Internet und die Civil Society als globale Informationsgesellschaft, in diesem Band.

Werle, Raymund/Leib, Volker (1999), The Internet Society and its Struggle for Recognition and Influence, erscheint in: Ronit, Karsten/Schneider, Volker (Hrsg.): Private Organisations and Global Politics, London, http://www.mpi-fg-koeln.mpg.de/publikation/working_papers/wp99-12/index.html.

Weyer, Johannes (1997), Konturen einer netzwerktheoretischen Techniksoziologie, in: Weyer u. a.: Technik, die Gesellschaft schafft. Soziale Innovation als Ort der Technikgenese. Berlin: 23-52.

Weyer, Johannes/Kirchner, Ulrich/Riedl, Lars/Schmidt, Johannes F. K. (1997), Technik, die Gesellschaft schafft. Soziale Innovation als Ort der Technikgenese. Berlin.