

Wenn Peers produzieren

Von Freier Software zu Freier Hardware und darüber hinaus

Christian Siefkes

Dieser Artikel ist in der *FifF-Kommunikation 1/2010* erschienen. Er wird unter den Bedingungen der *Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen Deutschland-Lizenz (CC-BY-SA) 3.0* veröffentlicht. Den Text der Lizenz erhalten Sie unter der URL <http://creativecommons.org/licenses/by-sa/3.0/de/> oder indem Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA schreiben.

Das Beispiel Linux

1991 hatte der junge finnische Informatikstudent Linus Torvalds eine verblüffende Idee: er begann damit, auf seinem neuerworbenen PC ein Betriebssystem zu schreiben. Zunächst ging es ihm nur darum, einige fehlende Funktionen für seinen Rechner nachzurüsten, doch nach einigen Monaten Bastelei bemerkte er, dass er ein System entwickelt hatte, das auch für andere Leute nützlich werden könnte. Er kündigte seine Arbeit öffentlich im Internet an – „Ich arbeite an einem (freien) Betriebssystem (nur ein Hobby...)“ – und bat um Rückmeldungen, welche Eigenschaften sich die anderen von einem solchen System wünschten. Einige Wochen später stellte er die Software ins Internet, was es jedem ermöglichte, Torvalds' Code herunterzuladen, zu verwenden und (bei entsprechenden Programmierkenntnissen) auch den eigenen Bedürfnissen anzupassen.

Torvalds' Ankündigung stieß auf gewaltiges Interesse, denn die damals verbreiteten Betriebssysteme konnten entweder wenig (wie DOS) oder sie waren teuer. Außerdem wurden sie in Firmen entwickelt, auf die die Nutzer/innen keinen Einfluss hatten. Dass jemand ein Betriebssystem öffentlich per Internet entwickelte, dabei die Nutzer explizit um Rückmeldungen und Mitarbeit bat, und dann auch noch die Ergebnisse allen frei zur Verfügung stellte, war eine Sensation. Es dauerte nur zwei Jahre, bis über hundert Leute an Linus Torvalds' System mitarbeiteten, das in Anlehnung an seinen Schöpfer *Linux* getauft wurde. Damals hatte das von Richard Stallman initiierte *GNU-Projekt* bereits viele freie Betriebssystemkomponenten entwickelt – durch die Kombination mit dem von Torvalds geschriebenen Systemkern entstand daraus ein praktisch nutzbares und komplett freies Betriebssystem.

Heute gehört Linux zu den drei verbreitetsten Betriebssystemen – neben Windows und Mac OS – und wird von Millionen von Menschen verwendet. Noch beliebter als bei privaten Anwender/innen ist Linux bei Firmen, die es insbesondere für Server einsetzen, die dauerhaft und zuverlässig laufen müssen. Auch wo die Leistungsanforderungen besonders hoch sind,

ist das System verbreitet, so laufen sogar 89% der 500 schnellsten Supercomputer unter Linux (TOP500 2009).

Der Erfolg von Linux basiert zum einen darauf, dass die Software selbst – wie alle Freie Software – ein *Gemeingut* ist, das jede/r frei verwenden, den eigenen Bedürfnissen oder Vorstellungen gemäß weiterentwickeln (bei entsprechenden Kenntnissen) und auch an andere weitergeben darf. Die Freiheiten, die Freie Software zum Gemeingut machen, wurden dabei schon in den 1980er Jahren von Richard Stallman beschrieben und exemplarisch in der *GNU General Public License* (GPL) umgesetzt – der bis heute am meisten verwendeten Lizenz für Freie Software, die auch von Linux genutzt wird.

Entscheidend für den Erfolg von Linux ist aber die *Community*, die hinter dem Betriebssystem steht und seine Entwicklung koordiniert. Die offene, dezentrale und scheinbar chaotische Art und Weise, in der Torvalds und seine Mitstreiter/innen zusammenarbeiten, ist als „Basar“-Modell in die Softwaregeschichte eingegangen (Raymond 1999) – im Gegensatz zum hierarchischen, sorgfältig geplanten „Kathedralenstil“, der nicht nur den Bau mittelalterlicher Kathedralen, sondern auch einen Großteil der in Firmen entwickelten Software prägte.

Die Beteiligung an Freie-Software-Projekten wird oft mit der 90-9-1-Regel beschrieben: 90% benutzen das System nur, etwa 9% tragen gelegentlich etwas zu seiner Weiterentwicklung bei, und nur 1% beteiligt sich regelmäßig und intensiv. Der Aufstieg mancher Nutzer/innen zu sporadisch oder auch intensiv Beitragenden erfolgt per „Selbstausswahl“ – es gibt keine Beteiligungsverpflichtung, aber auch wenig Hindernisse. Jede/r sucht sich selbst aus, ob und wie viel sie oder er tun möchte. Oft beginnt die Beteiligung damit, dass jemand einen Bug findet und meldet – etwas, das nicht richtig funktioniert; eventuell schreibt sie dann auch einen Patch, einen „Softwareflicken“, um den Fehler zu beheben – und schon hat sie zur Weiterentwicklung der Software beigetragen. Man kann aber auch Patches schreiben, die die Funktionalität des Systems erweitern, und so beispielsweise Funktionen nachrüsten, die man selbst benötigt; oder man testet das System oder schreibt Dokumentation.

Der neu entstandene Code wird dann von erfahreneren Beteiligten begutachtet, ob er auch ins System aufgenommen werden kann, ohne etwas kaputt zu machen, und eventuell korrigiert. Bei Linux gibt es ungefähr hundert „Maintainer“, die die Verantwortung für bestimmte Teilsysteme übernommen haben und dafür sorgen, dass bei ihrer Weiterentwicklung alles glatt läuft. Im Zentrum des Ganzen steht nach wie vor Linus Torvalds selbst, der aber nur noch einen kleinen Teil der Entscheidungen selber trifft. (In manchen anderen Projekten wie dem alternativen freien Betriebssystem FreeBSD gibt es dagegen ein Kernteam von mehreren Hauptverantwortlichen, das von den Projektbeteiligten regelmäßig neu gewählt wird.)

Je länger und intensiver jemand mitarbeitet, desto schneller und unproblematischer werden seine Änderungen von den zuständigen Maintainern übernommen – bis man schließlich vielleicht selbst Maintainer wird. Maintainer zu werden bringt Verantwortung und zugleich mehr Einfluss auf die weitere Entwicklung des Projekts, aber es bringt keine Macht über andere. Vielmehr sind die Maintainer stets auf die freiwilligen Beiträge der anderen angewiesen. Sie können andere davon abhalten, dem Projekt zu schaden, indem sie die Aufnahme von schlechtem Code in die Software verweigern, aber sie können niemand dazu zwingen, etwas Bestimmtes zu tun.

Generell gibt es für den Entwicklungsprozess Richtlinien (Empfehlungen), aber kaum strikte Regeln. Wer gegen die üblichen Praktiken verstößt, muss damit rechnen, von den anderen Entwickler/innen per E-Mail „geflamed“, d.h. beschimpft und in rauen Worten auf

den Verstoß hingewiesen zu werden – etwas Schlimmeres passiert selten, härtere Sanktionen wie der Ausschluss aus dem Projekt kommen kaum vor.

Gemeingüter, Beiträge und freie Kooperation

Die Erfolgsgeschichte von Linux zeigt exemplarisch einige Merkmale auf, die für *Peer-Produktion* wesentlich sind. Ein erstes Merkmal ist, dass *Gemeingüter* (engl. *Commons*) von großer Bedeutung sind. Gemeingüter sind Ressourcen und Güter, die allen zustehen und die gemäß selbstdefinierten Regeln gemeinsam oder anteilig genutzt werden. Freie Software ist ein solches Gemeingut (wir werden später noch weitere kennenlernen), und zu den Regeln, die sich die Freie-Software-Community gegeben hat, gehören insbesondere die „vier Freiheiten“, die Freie Software ausmachen. Die Freiheit, erstens das Programm für jeden Zweck einsetzen und es zweitens den eigenen Bedürfnissen anpassen zu dürfen, es drittens an andere weitergeben und es viertens verbessern und diese Verbesserungen allen zugänglich machen zu dürfen (GNU-Projekt 2002).

Stallmans GPL – die beliebteste Lizenz für Freie Software – fordert dabei, dass diese vier Freiheiten auch für jede *abgeänderte* Version der Software erhalten bleiben – wenn ich GPL-Software bearbeite und die geänderte Version veröffentliche, muss ich sie selbst unter GPL stellen. Dieses Prinzip wird „Copyleft“ genannt, da es sich das normale Copyright bzw. Urheberrecht zunutze macht, um dessen Funktion quasi „umzudrehen“. Während das normale Urheberrecht die Autor/innen zu nichts verpflichtet und den Nutzer/innen fast nichts erlaubt, macht das Copyleft das Gegenteil: es *erlaubt* den Nutzern sehr viel, indem es ihnen die genannten vier Freiheiten gewährt, und *verpflichtet* zugleich alle künftigen Autoren, den Nutzern ihrer eigenen verbesserten Versionen dieselben Rechte einzuräumen. Regelungen wie das Copyleft stehen somit nicht im Widerspruch zur Freiheit der Beteiligten, sondern stärken diese, indem sie Schutz und Ausbau der Commons ermöglichen.

Peer-Produktion baut auf Gemeingütern auf, und sie erzeugt neue Gemeingüter oder pflegt und verbessert die vorhandenen. Andere Ressourcen wie die verwendeten Computer sind bei Peer-Projekten typischerweise in Privatbesitz, aber sie werden zum Erreichen der Projektziele benutzt, nicht zum Erzielen eines finanziellen Gewinns. Sie fungieren als *Besitz* (etwas, das man benutzt), nicht als *Eigentum* (etwas, das man verkaufen oder verwerten kann). *Peer-Produktion basiert somit auf Gemeingütern und Besitz (nicht auf Eigentum)*.

Im Gegensatz zur Produktion für den Markt findet die Peer-Produktion nicht für den Verkauf, sondern für den Gebrauch statt. Peer-Projekte haben ein gemeinsames Ziel, und alle Teilnehmer/innen tragen auf die eine oder andere Weise etwas zu diesem Ziel bei. Und zwar in einem Großteil der Fälle nicht, um damit Geld zu verdienen, sondern weil sie die Ziele des Projekts teilen und wollen, dass es erfolgreich ist; weil sie genießen, was sie da tun; oder weil sie der Community etwas „zurückgeben“ wollen. Bei marktwirtschaftlichen Aktivitäten wird etwas eingetauscht, zumeist eine Ware gegen Geld – Angestellte verkaufen eine bestimmte Ware, ihre Arbeitskraft, und Firmen verkaufen die Waren, die sie produzieren. Im Gegensatz dazu *basiert Peer-Produktion nicht auf Tausch, sondern auf den Beiträgen* der Beteiligten.

Wenn Firmen sich an Peer-Projekten beteiligen (was bei Freier Software häufig der Fall ist), ist es um die Motivation natürlich anders bestellt – jede Firma (bzw. ihre Inhaber- und Leiter/innen) will und muss Geld verdienen, d.h. das zuvor investierte Geld in mehr Geld verwandeln, und jede Aktivität von Firmen lässt sich direkt oder indirekt durch dieses universelle Motiv erklären. Trotzdem sind viele Firmen, wenn sie Freie Software unterstützen, unmittelbar

am *Gebrauchswert* und nicht am *Tauschwert* (finanziellen Wert) der Software interessiert – viele Firmen unterstützen etwa die Entwicklung des Webservers *Apache* oder von *Linux*, weil sie diese Systeme auf ihren eigenen Servern laufen lassen wollen. Sie sind am Gebrauch der Software interessiert, genau wie Privatpersonen, die eine bestimmte Software nutzen wollen und deshalb dazu beitragen, dass sie besser wird. Bei Firmen steht hinter diesem Interesse am Gebrauchswert natürlich das ultimative Interesse, Geld zu verdienen (und deshalb z.B. *Apache* zu verwenden, weil er besser und/oder billiger ist als unfreie Alternativen). In anderen (und selteneren) Fällen geht es Firmen nicht selbst um den Gebrauchswert der Software, sondern direkt ums Geldverdienen – sie verkaufen z.B. Support für Freie Software, was nur bzw. besser funktioniert, wenn sie selbst zur Weiterentwicklung der Software beitragen.

Dass Firmen am Geldverdienen interessiert sind, versteht sich von selbst. Interessanter sind zwei anderen Faktoren: zum einen, dass die vielen Beteiligten, die *nicht* durch eine Firma bezahlt werden, sich aus ganz anderen Motiven beteiligen – dass es ihnen zumeist *nicht* ums Geldverdienen oder die Steigerung des eigenen Marktwerts geht; und zum anderen, dass die so entstehenden Projekte so erfolgreich sind, dass es für Firmen Sinn macht, sich zu beteiligen. Das zeigt, dass diese selbstorganisierte, nicht in erster Linie durch Firmen oder den Staat geprägte Produktionsweise viel mehr als eine bloße Hobby- oder Freizeitaktivität ist – sie ist so erfolgreich, dass Firmen es sich nicht mehr leisten können, sie zu ignorieren (selbst *Microsoft* beteiligt sich inzwischen an *Open-Source*-Projekten, nach langjährigem Boykott und *Anti-GPL*-Propaganda).

Anders als bei Firmen und planwirtschaftlichen Systemen gibt es innerhalb von *Peer*-Projekten keine Befehlsstrukturen. Das heißt keineswegs, dass die Projekte unstrukturiert wären (wie bei *Linux* gibt es bei den meisten Projekten *Maintainer* oder *Administrator*/innen, die das Projekt auf Kurs halten und entscheiden, ob Beiträge integriert oder zurückgewiesen werden), aber niemand kann anderen befehlen, etwas zu tun, und niemand ist gezwungen, anderen zu gehorchen. Auch *Maintainern* bleibt nur, die Beteiligten davon zu überzeugen, dass eine bestimmte Aktivität sinnvoll ist – anordnen können sie nichts.

Der unübersetzbare Begriff der „*Peers*“ bezieht sich auf diese *freiwillige, ungezwungene Kooperation zwischen Gleichberechtigten, die sich niemand unterordnen müssen*. Die Strukturen und Organisationsformen von Projekten entwickeln sich dabei gemäß den Bedürfnissen und Vorstellungen der Beteiligten – in einem offenen, niemals abgeschlossenen Prozess entwickeln die Projekte die Regeln und Organisationsformen, die zum Erreichen ihrer Ziele am besten geeignet sind.

Freie Software und *Peer*-Produktion sind nicht dasselbe. Einerseits gibt es *Peer*-Produktion in vielen weiteren Bereichen (Beispiele folgen im nächsten Absatz), andererseits erfordert *Peer*-Produktion eine *Community* voneinander unabhängiger Beteiligter. Wenn nur eine einzelne Autor/in oder eine einzelne Firma eine Freie Software entwickelt, handelt es sich nicht im *Peer*-Produktion im engeren Sinne. *Potenziell* allerdings schon, sind die Freien Lizenzen doch immer eine Einladung zur Beteiligung durch andere – früher oder später können andere die Software entdecken und sie für eigene Zwecke anpassen oder weiterentwickeln, was die Möglichkeit eröffnet, dass aus dem Privatprojekt eine *Community* wird.

Die Vielfalt der *Peer*-Produktion

Was bei *Linux* funktionierte, wird in ähnlicher Form mittlerweile in unzähligen anderen Projekten praktiziert. Ein weiterer kolossaler Erfolg ist die 2001 gegründete *Wikipedia*, die

freie Enzyklopädie, an der jede/r mitarbeiten kann und deren deutsche Ausgabe schon mehr als eine Million Artikel umfasst. Linux und die Wikipedia stehen exemplarisch für zwei Communities – die *Freie-Software-Bewegung* (auch als *Open-Source-Bewegung* bekannt) und die *Freie-Kultur-Szene* –, die aber natürlich viel größer sind als ihre jeweiligen Flaggschiffe. Es gibt Hunderttausende Freier Softwareprogramme und Millionen von Werken (meist Texte, Bilder, Musik, seltener Filme), die unter „Creative Commons“-Lizenzen veröffentlicht werden, den bekanntesten Lizenzen für Freie Kultur.

Daneben existieren viele weitere dezentrale Peer-Communities, die sich um die Schaffung und Bewahrung eines Gemeinguts organisieren. So ist es Anliegen der *Open-Access-Community*, durch die Schaffung von freiem Zugang zu wissenschaftlichen Veröffentlichungen und Experimentaldaten das wissenschaftliche Wissen wieder in das Gemeingut zu verwandeln, das es traditionell war. *Freie Funknetze* sind selbstorganisierte Computernetzwerke, die freien Datenverkehr zwischen Computern ermöglichen und freie Zugangspunkte ins Internet zur Verfügung stellen.

Interkulturelle Gärten („community gardens“) sind kleine selbstverwaltete Gemeingüter, die an vielen Orten der Welt, meist in städtischen Umgebungen, entstanden sind. Diese Gärten bedeuten den Menschen, die sie hegen oder besuchen, eine Verbindung zur Natur und zu einer aktiven Gemeinschaft. Die Beteiligten der *BookCrossing-Community* lassen Bücher, die sie nicht mehr brauchen, weiter „wandern“, gemäß der Idee, dass Bücher geschrieben werden, um gelesen zu werden und nicht in Regalen zu verstauben. Und die Mitglieder des *CouchSurfing-Netzwerks* bieten Reisenden unentgeltlich einen Schlafplatz in der eigenen Wohnung oder eine Stadtführung an – über eine Million Menschen in über 200 Ländern sind schon dabei.

Die Zukunft der Peer-Produktion

Im Bereich der Informationsgüter führt die Peer-Produktion bereits zu Ergebnissen, die der herkömmlichen Produktion durch Firmen ebenbürtig oder überlegen sind. Die Wikipedia hat herkömmlichen Konkurrenten wie dem Brockhaus und der Encyclopaedia Britannica wohl weitgehend den Garaus gemacht, Linux ist eines der erfolgreichsten Betriebssysteme und im High-End-Bereich (wie wir gesehen haben) schon absolut führend, der Apache-Webserver ist seit vielen Jahren der am häufigsten genutzte Webserver im Internet, und der Webbrowser Firefox scheint seinen schärfsten Konkurrenten – den MS Internet Explorer – in Deutschland mittlerweile überholt zu haben. In anderen Bereichen wird Peer-Produktion ebenfalls erfolgreich praktiziert (siehe oben), doch gibt es hier noch keine so durchschlagenden Erfolgsgeschichten.

Wir sollten aber nicht vergessen, dass die hier beschriebenen Entwicklungen alle sehr neu sind – Linux und Apache sind noch keine 20 Jahre alt, die Wikipedia gibt es seit weniger als zehn Jahren. Wenn die 90er Jahre den Durchbruch der Freien Software brachten und die 00er Jahre die explosionsartige Ausbreitung der Freien Inhalte – Wikipedia und Creative Commons –, dann könnten die 10er Jahre die große Zeit der Peer-Produktion jenseits reiner Informationsgüter einläuten. Denn in dieser Hinsicht tut sich derzeit eine Menge, Stichworte dafür sind insbesondere „Freies Design“ und „Community-basierte Infrastrukturen“.

Freies Design

In der Peer-Community wird Wissen ganz selbstverständlich als Gemeingut behandelt. Dass dies für alles öffentlich relevante Wissen gelten sollte, drückt die Wikimedia-Stiftung (die hinter der Wikipedia steht) so aus: „Imagine a world in which every single human being can freely share in the sum of all knowledge.“ – „Stellen Sie sich eine Welt vor, in der jeder Mensch freien Zugang zur Gesamtheit allen Wissens hat.“ (Wikimedia Foundation 2010).

Bislang ging es um zwei Arten Freien Wissens: Freie Software und Freie Inhalte. Derzeit ist eine weitere Art von Freiem Wissen im Kommen: *Freies Design*, auch *Open-Source-Hardware* oder (etwas ungenau) *Freie Hardware* genannt. Gemeint sind Projekte, die gemeinsam materielle Produkte entwerfen und dabei Objektbeschreibungen, Konstruktionspläne und Materiallisten öffentlich zur Verfügung stellen und als Gemeingut teilen. Das US-amerikanische Magazin *Make* veröffentlicht jährlich einen großen Report zum Thema, der in der Ausgabe von Ende 2009 schon über 125 Projekte enthielt – mehr als doppelt so viele wie im Vorjahr (*Make* 2009). Dazu gehören Plattformen für Computerhardware wie *Arduino*, *Bug Labs* und *OpenCores*, Telekommunikations-Hardware wie *Asterisk* und *Openmoko* sowie Geräte für Musik und Kunst (z.B. ein Synthesizer und diverse MP3-Player). Es gibt auch medizinische Projekte, wie das *Open Prosthetics Project*, das frei nutzbare Prothesen entwickelt.

Besonders interessant sind Dinge, die andere Dinge produzieren können – dazu gehört der *Contraptor*, eine Plattform für den experimentellen Bau von CNC- und anderen Maschinen, sowie sogenannte 3D-Drucker wie *Fab@Home* und *RepRap*. CNC-Maschinen stellen Objekte aus Holz oder Metall her, indem sie die nicht benötigten Teile eines Materialblocks absägen oder wegfräsen (subtraktive Fertigung). 3D-Drucker produzieren Gegenstände aus Plastik, indem sie das in Pulverform gebrachte Material Schicht für Schicht auftragen, wobei die einzelnen Schichten ähnlich wie bei einem Tintenstrahldrucker quasi „ausgedruckt“ werden (additive Fertigung). Solche Geräte können Grundlage einer freien Produktionsmittel-Infrastruktur werden.

Bei Freien Designs kann man wie bei Freier Software und anderen Freien Projekten darauf bauen, dass die Offenheit zu großer Vielfalt führt, da jede/r mitmachen und eigene Anpassungen oder Erweiterungen beitragen kann. Dies erhöht die Chancen, dass eine dem eigenen Geschmack oder Bedürfnissen entsprechende Variante bereits von anderen entworfen und online gestellt wurde. So können auch Spezialbedürfnisse abgedeckt werden, für die kein „Markt“ vorhanden ist und die von kommerziellen Anbietern normalerweise ignoriert werden.

Community-basierte Infrastrukturen

Designs und Baupläne nutzen nicht allzu viel, wenn es am Zugang zu den benötigten Produktionsmitteln und Ressourcen mangelt. Zumindest im Bereich der Produktionsmittel ist dabei einiges in Bewegung geraten, da viele Produktionstechniken im Zuge der technologischen Entwicklung günstiger und zugänglicher werden. Heute können Hobbyist/innen und Peer-Projekte mit günstig erworbenen oder selbstgebauten Maschinen produzieren, wofür noch vor wenigen Jahrzehnten eine kapital- und personalintensive Fabrik nötig gewesen wäre. Dazu gehören die schon erwähnten CNC-Maschinen und 3D-Drucker, aber auch für andere Techniken wie das zur Produktion der meisten Plastikprodukte eingesetzte *Spritzgießen* finden sich Selberbau-Projekte im Internet.

Natürlich kann nicht jede/r alle benötigten Tools im eigenen Keller haben. Wichtig sind daher Projekte, in denen sich z.B. Einwohner/innen eines Dorfs oder Stadtteils zusammenschließen, um gemeinschaftlich die benötigte Infrastruktur aufzubauen und zu betreiben. Einige Beispiele dafür gibt es schon, so haben die Einwohner der südafrikanischen Gemeinde Scarborough mit der *Scarborough Wireless User Group* ein dezentrales Mesh-Netzwerk eingerichtet, das ihrer Stadt den Zugang zum Internet und Telefonnetz ermöglicht. Die dafür nötige Hard- und Software stammt aus zwei Freies-Design-Projekten, dem *Village Telco* und dem *Free Telephony Project*. Die benötigten WLAN-Router werden jeweils von Bürger/innen gekauft und dem Netz zur Verfügung gestellt – es gibt niemand, dem das ganze Netz oder ein Großteil davon gehören würde. Zudem gibt es freiwillige Abgebühren, um die extern anfallenden Kosten (etwa für die DSL-Zugänge zum eigentlichen Internet) zu decken. Wer sich nicht finanziell beteiligen will oder kann, kann das Netzwerk trotzdem nutzen, allerdings werden die Verbindungen von (finanziell) Beitragenden im Zweifelsfall vorrangig bedient (Rowe 2010). Auf diese Weise kann sich das Netzwerk selber tragen, ohne auf einzelne Wohltäter/innen oder externe Geldgeber angewiesen zu sein, doch gleichzeitig ist sichergestellt, dass niemand ausgeschlossen wird.

Ähnliche Entwicklungen sind im Bereich offener, auf Freie Hard- und Software aufbauender Produktionsstätten zu erwarten. Ein Vorläufer, der allerdings noch größtenteils auf nichtfreie Produktionsmaschinen zurückgreift, sind die *Fab Labs* (<http://fab.cba.mit.edu/>), die in den letzten Jahren in vielen Ländern (seit kurzem gibt es auch eines in Aachen) entstanden sind. Fab Labs sind offene Werkstätten, die den Anspruch haben, „beinahe alles“ produzieren zu können. Ganz so weit ist es noch nicht, doch allerhand nützliche Dinge (z.B. Mobiliar u.a. Holzgegenstände, Kleidung, Platinen u.a. Computerzubehör) lassen sich dort bereits herstellen. Die größte Beschränkung ist bislang, dass die verwendeten Werkzeuge proprietär sind – sie müssen bei bestimmten Herstellern eingekauft werden, ihr Design ist nicht offen gelegt, und niemand kann bzw. darf sie einfach nachbauen und verändern. Doch gibt es in der Community Anstrengungen, diese Abhängigkeit zu überwinden und die „produktive Rekursion“ zu erreichen: ein Netzwerk Freier Produktionsstätten, deren Ausstattung zu 100% Freies Design ist und in den zusammenarbeitenden Werkstätten selbst reproduziert werden kann. Dies würde es ermöglichen, weitere Werkstätten aufzubauen, ohne die benötigte Ausstattung kaufen zu müssen.

Dadurch würde sich die Abhängigkeit vom Markt spürbar reduzieren, auch wenn damit selbstredend noch nicht alle Probleme gelöst sind. Offen bleibt insbesondere die Frage, wer die benötigten natürlichen Ressourcen kontrolliert, denn solange sich die allermeisten Ressourcen im Privatbesitz einzelner Personen oder Konzerne befinden und gegen Geld erworben werden müssen, bleiben die Möglichkeiten einer generell „freien“, geldlosen Produktionsweise stark beschränkt.

Reclaim the Commons!

Die Commons-Perspektive kann hier Antworten bieten, denn die gemeinsame Verwaltung und Nutzung natürlicher Ressourcen – etwa Land, Wasser, Luft und Wälder – als Gemeingüter war und ist an vielen Orten und auf vielfältige Weise gängige Praxis. Commons-Forscher/innen wie Elinor Ostrom (1999, 2009), die für ihre Arbeiten mit dem Wirtschaftsnobelpreis 2009 ausgezeichnet wurde, und Historiker/innen wie Peter Linebaugh (2008) haben dazu viel zu sagen. Die Gemeinschaften der „Commoners“, die diese Ressourcen nutzten und sich

um sie kümmern, hatten Regeln entwickelt, um allen fairen Zugang zu ermöglichen und zugleich Übernutzung und Zerstörung der Commons zu verhindern.

Dagegen ist bei privatisierten und „eingezäunten“ ehemaligen Gemeingütern maximale Ausbeutung ohne Rücksicht auf andere und oft sogar ohne Rücksicht auf die Zukunft ganz normal – Bodenerosion, das Schrumpfen der Regenwälder, der drohende Kollaps von Fischbeständen, die für Menschen und Natur oft verheerenden Konsequenzen der Erdölförderung etwa in Südamerika (Klima-Bündnis 2004), und nicht zuletzt die Übernutzung der Atmosphäre als „Emissionshalde“ und die dadurch ausgelöste, für die Zukunft der Menschheit bedrohliche globale Erwärmung sprechen hier eine deutliche Sprache. Es gab und gibt zahlreiche Auseinandersetzungen, in denen sich Menschen gegen diese Zumutungen wehren und sich um eine Wiederaneignung der Commons bemühen (Bollier 2002, Helfrich 2010), und noch viel mehr werden nötig sein.

In einer auf Peer-Produktion basierenden Gesellschaft werden das Wissen und die Natur Gemeingüter sein, die allen zustehen, und die Menschen werden sich per „Selbstausswahl“ die Aufgaben aussuchen, mit denen sie sich beschäftigen, nach dem Motto: „Was man gerne macht, macht man meist auch gut“. Noch ist es nicht so weit, doch der Weg dorthin zeichnet sich schon ab. Wer mehr wissen möchte, kann in Siefkes (2008, 2009) sowie bei www.keimform.de weiterlesen.

Literatur

- Bollier, David (2002): Reclaiming the Commons. In: Boston Review 27 (3–4). URL <http://www.bostonreview.net/BR27.3/bollier.html> (Zugriff am 27.1.2010).
- GNU-Projekt (2002): Die Definition Freier Software. URL <http://www.gnu.org/philosophy/free-sw.de.html> (Zugriff am 15.1.2010).
- Helfrich, Silke (2010): Commonsblog. URL <http://commonsblog.wordpress.com/> (Zugriff am 27.1.2010).
- Klima-Bündnis (2004): Der Fluch des Erdöls. URL <http://www.erdoelinamazonien.org/fluch.html> (Zugriff am 27.1.2010).
- Linebaugh, Peter (2008): The Magna Carta Manifesto. Berkeley: University of California Press.
- Make (2009): Open Source Hardware 2009. URL http://blog.makezine.com/archive/2009/12/open_source_hardware_2009_-_the_def.html (Zugriff am 25.1.2010).
- Ostrom, Elinor (1999): Die Verfassung der Allmende. Tübingen: Mohr.
- Ostrom, Elinor (2009): Gemeingütermanagement – eine Perspektive für bürgerschaftliches Engagement. In: Silke Helfrich/Heinrich-Böll-Stiftung (Hrsg): Wem gehört die Welt? München: Oekom, S. 218–228. URL <http://commonsblog.wordpress.com/das-buch-el-libro/> (25.1.2010).
- Raymond, Eric (1999): Die Kathedrale und der Basar. URL <http://gnuwin.epfl.ch/articles/de/Kathedrale/> (Zugriff am 14.1.2010).
- Rowe, David (2010): Baboons, Mesh Networks, and Community. URL <http://www.rowetel.com/blog/?p=124> (Zugriff am 22.1.2010).
- Siefkes, Christian (2008): Beitragen statt tauschen. Neu-Ulm: AG SPAK Bücher.

Originalausgabe: From Exchange to Contributions. Berlin 2007. URL:
<http://www.peerconomy.org/> (Zugriff am 25.1.2010).

- Siefkes, Christian (2009): Ist Commonismus Kommunismus? In: PROKLA 39 (2), S. 249–268. URL:
<http://www.keimform.de/2009/07/17/ist-commonismus-kommunismus-html/>
(Zugriff am 25.1.2010).
- TOP500 (2009): Operating System Family Share for 11/2009. URL
<http://www.top500.org/stats/list/34/osfam> (Zugriff am 14.1.2010).
- Wikimedia Foundation (2010): Strategic Planning. URL
<http://strategy.wikimedia.org/> (Zugriff am 23.1.2010).