

Auszug aus Streifzüge bei Context XXI

(<http://contextxxi.org/freie-software-und-kapitalismus.html>)

erstellt am: 29. März 2024

Datum dieses Beitrags: März 2002

Freie Software und Kapitalismus*

■ MICHAEL HEINRICH ■ SABINE NUSS

Freie Software als „Anomalie“

Die Herausbildung weltweiter Verknüpfung von Computern und Computernetzen stellt nicht nur selbst eine neue Kommunikationstechnologie dar, sondern bringt auch sukzessive neue Informationsprodukte, Produktions- und Distributionsformen hervor. Musik, Literatur, Software oder Filme gab es zwar schon vor dem Internet, aber mittels der Digitalisierung können die „Netzinhalte“ (Text, Klang, Bild, Algorithmen) als Bits ohne größeren Aufwand und verlustfrei (das Original „leidet“ nicht) vervielfältigt und direkt an die computerisierten Nutzer verbreitet werden. Ein Zwischenhändler oder ein Distributor, der zugleich das Monopol auf die Verwertungsrechte der Inhalte hat, wird damit obsolet. So kann ein Musiker sein neues Stück direkt ins Netz stellen, ein Schriftsteller seinen Text, ein Programmierer seine Software. Weiterhin kann an einem Informationsprodukt, wie beispielsweise Software, weltweit und dezentral gearbeitet werden, vorausgesetzt die technische Infrastruktur und das know how sind gegeben.

Dieses Charakteristikum des Internet (Digitalisierung + weltweiter Transfer) stellt die Frage nach den kapitalistischen Verwertungsmöglichkeiten von Informationsprodukten neu. Zur Verwertung benötigt man nicht nur ein Produkt, welches auf ein zahlungswilliges und zahlungsfähiges Bedürfnis stößt, sondern auch die Gewissheit, dass dieses Produkt nirgendwo anders ohne großen Aufwand quasi umsonst zu haben ist: Verwertung braucht Knapp-

heit. Das Internet bietet jedoch zumindest im gegenwärtigen Zustand einen „Überfluss“ an Daten, oder besser: die Abschliessung von Daten im Netz unter dem exklusiven Titel des privaten Eigentums ist technisch noch nicht umfassend realisiert. So schwirren und flirren von Rechner zu Rechner quer über Ländergrenzen hinweg Informationsprodukte in Form von Bits und Bytes und tun so, als hätten sie mit der Warenform nichts am Hut.

Während Unternehmen versuchen, die traditionellen Intellectual Property Rights (IPR) mittels technischer Sicherungsinstrumente (Kopierschutz) auch im Cyberland durchzusetzen und damit eine Art virtueller Enclosures [1] (Aneignung von Datenland) initiieren (vgl. dazu Nuss 2002), während zugleich die Staaten Nutzer und Anbieter von frei zugänglichen digitalen Inhalten illegalisieren, um auch das Cyberland der herrschenden Rechtsordnung zu unterwerfen, entstehen andererseits Produkte und Produktionsformen, die sich diesen Enclosures und der Verwertung zu entziehen meinen: *Die Freie Software*.

Freie Software ist Software, die sich von kommerzieller Software im wesentlichen dadurch unterscheidet, dass ihr Quellcode [2] bei Weitergabe des Programms mitgeliefert, einsehbar und veränderbar ist. Die Nutzung Freier Software ist kostenlos (das heisst nicht, dass ihre Träger, wie z. B. CD-ROM, kostenlos sind). Die Produktion Freier Software erfolgt im wesentlichen dezentral (via Internet, örtlich ungebunden) von unterschiedlich organisierten Entwicklerteams und zumeist ohne finanzielle Entlohnung. Die Entwickler

befinden sich mit dieser Tätigkeit (in der Regel) in keinen eigens dafür zweckbestimmten Lohnarbeitszusammenhängen, sondern programmieren in der Freizeit, während der regulären Arbeitszeit oder nach Feierabend; und bei Studenten im Rahmen oder während des Studiums. Der Zweck der Programmierarbeit ist nicht der Tauschwert, sondern der Gebrauchswert: es wird gearbeitet, um ein Produkt herzustellen, nicht um zu tauschen. Das Produkt, die Freie Software, wird ins Netz gestellt; zum einen kann so jeder Programmierer weltweit an der Entwicklung mitarbeiten, zum anderen können Anwender sich so ebenfalls weltweit das Produkt auf ihre Festplatte laden – ohne dafür zu bezahlen.

Privateigentum an Freier Software gibt es nicht. Das bekannteste rechtliche Instrumentarium, welches Freie Software vor einer Reprivatisierung schützen soll, ist die GNU General Public License (GPL) – auch „Copyleft“ genannt. [3] Neben GPL entstanden und entstehen allerdings zahlreiche Lizenzen, die die strikten Regelungen der GPL aufweichen (vgl. ausführlich zu Freier Software: Grassmuck 2000). Zu den bekanntesten Freien Software Programmen zählen u. a. das Betriebssystem Gnu/Linux, der Internet-Server Apache, der Compiler Gcc, die Programmiersprache Perl, das Desktop KDE, das Bildbearbeitungsprogramm Gimp.

Auf neue technische Entwicklungen folgen theoretische Reflexionen, die das Neue einordnen und interpretieren. Bei der Freien Software sind die reflektierenden Wissenschaftler, Politiker, Künstler, Journalisten, Techniker usw. mit einer „Anomalie“ konfrontiert, wie

sie – in ganz anderen Zusammenhängen – von dem Wissenschaftshistoriker Thomas Kuhn (1962) beschrieben wurde. Kuhn betonte, dass die wissenschaftliche Entwicklung nicht einfach in einer Abfolge bestätigter oder falsifizierter Theorien besteht, sondern dass der Theoriebildung „Paradigmen“ zugrunde liegen, Muster, nach denen Theorien überhaupt gebildet werden und in denen sich eine bestimmte Weltanschauung niederschlägt. Paradigmen können durch empirische Ergebnisse nicht ohne weiteres widerlegt werden, da sie die Verarbeitung dieser Ergebnisse ganz wesentlich organisieren. Eine „Anomalie“ ist ein Phänomen, das sich der üblichen Verarbeitung entzieht, es passt nicht so richtig in das herrschende Paradigma hinein. Kuhn beschreibt verschiedene Arten, wie die wissenschaftliche Community auf solche Anomalien reagieren kann. Die Anomalie kann von vorneherein ignoriert werden (mitunter auch, weil sie gar nicht als Anomalie wahrgenommen wird). Wird sie wahrgenommen, dann wird normalerweise versucht, sie mit Hilfe zusätzlicher Annahmen doch irgendwie in das herrschende Paradigma zu integrieren. Wenn dies misslingt, wird sie meistens als „ungelöstes Problem“ zur Seite gelegt, in der Hoffnung, dass künftige Forscher das Problem lösen werden. Ein besonderer Fall liegt jedoch vor, wenn diese Anomalie in ein neues Paradigma passt, das mit dem herrschenden Paradigma konkurriert. Dann wird die Anomalie von den Anhängern des neuen Paradigmas begierig als „Bestätigung“ aufgegriffen.

Zu den paradigmatischen Grundlagen der bürgerlichen Ökonomie gehört die Auffassung, dass nur Privateigentum für die Einzelnen eine ausreichende Anreizfunktion habe und daher auch nur Privateigentum zu (wirtschaftlicher) Effizienz führen könne. Dieser Gedanke hat eine lange Tradition. Schon Aristoteles machte ihn gegen die Staatsutopie Platons (die für die herrschende Klasse eine auf Gemeineigentum beruhende Gemeinschaft vorsah) geltend. Mit John Locke (1632-1704) – der zentralen Gestalt der frühen bürgerlichen Sozialphilosophie – erfährt dieser Gedanke dann die für die bürgerliche Ökonomie entscheidende Zuspitzung. Locke betrachtete die „eigene Arbeit“ (zu der er immer schon – auch im „Naturzustand“

– die Arbeit des eigenen Knechtes rechnet) als Grundlage des Eigentums: was ich mit eigener Arbeit „vermischen“ kann, wird zu meinem Eigentum. Gemeineigentum ist automatisch unproduktiv, denn wenn es wirklich Gemeineigentum bleiben soll, darf es niemand bearbeiten. Privateigentum, Arbeit und Produktivität wird damit sowohl funktional (produktive Arbeit ist nur möglich auf der Grundlage von Privateigentumsverhältnissen) wie auch normativ (Arbeit legitimiert Eigentum) zusammengeschlossen. Im 20. Jahrhundert wurde dieser Gedanke von Douglass North, der nicht zuletzt dafür den Nobelpreis für Wirtschaftswissenschaften erhielt, „modernisiert“ ausformuliert. Die Kernthese von North lautet: „Je gesicherter die (privaten) Eigentumsrechte, desto größer der Anreiz, desto effizienter die Wirtschaftsleistung“ und damit versuchte er den Verlauf der Wirtschaftsgeschichte von der Jungsteinzeit bis zur Gegenwart zu erklären.

Die Freie Software steht jedoch quer zu diesen Auffassungen: nicht nur verzichten ihre Entwickler auf ihre Eigentumsrechte (ein solcher Verzicht ist auch in anderen Bereichen, etwa bei ehrenamtlichen Tätigkeiten nicht unbekannt), dieser Verzicht auf die Eigentumsrechte ist hier jedoch die Vorbedingung einer weltweiten Kooperation, durch die das Produkt überhaupt hervorgebracht und verbessert wird. Ohne Privateigentum existieren hier sowohl Arbeitsanreiz als auch Effizienz. Freie Software bricht den normativen wie auch den funktionalen Zusammenhang von Arbeit, Eigentum und Effizienz auf. Insofern ist Freie Software eine „Anomalie“ für das Privateigentumsparadigma der bürgerlichen Ökonomie.

„Keimform-Theoretiker“

Von der herrschenden, bürgerlichen Ökonomie wird die Anomalie „Freie Software“ schlicht ignoriert. Kritiker des herrschenden Paradigmas haben sich dagegen ausführlich mit Freier Software beschäftigt. Wir wollen uns im folgenden mit den „Keimform-Theoretikern“, beschäftigen, die meinen in der „Anomalie“ Freie Software einen subversiven Gehalt zu entdecken: Die Freie Software wird als „Chance“ gesehen, den Kapitalismus zu überwinden. Freie Software fungiert in diesem Verständ-

nis als „Keimform“ einer künftigen nicht-kapitalistischen Gesellschaft.

Dass Freie Software Arbeitsanreize schafft und effizient ist, wird von den „Keimform-Theoretikern“ mit ihren spezifischen Produktionsbedingungen erklärt: Die Art und Weise, wie Freie Software entwickelt wird, sei eine im Gegensatz zur kapitalistisch organisierten Lohnarbeit nicht entfremdete Arbeit. Die Abwesenheit von Zwang (Verwertungszwang, Konkurrenzdruck, usw.) führe zu individueller Selbstentfaltung: Spass und Lust an der Tätigkeit und das Interesse an der Nützlichkeit des Produkts (nicht am Tauschwert) seien der treibende Motor und die Motivation der (in der Regel unbezahlten) Programmierung von Freier Software. „Nur in freien Projekten, in denen sich Einzelne nicht wie in Kommerzprojekten nur auf Kosten anderer durchsetzen können, sondern nur in Kooperation mit ihnen, steht das eigene Interesse nicht im Widerspruch zu den Interessen anderer. Diese Form der eigenen Entfaltung in einer kooperativen Form meine ich mit dem Begriff der Selbstentfaltung“ (Meretz 2000: 9). Die Möglichkeit der Selbstentfaltung in einem anspruchsvollen Tätigkeitsbereich schafft einerseits den Arbeitsanreiz, andererseits soll gerade diese vom Konkurrenzdruck freie Selbstentfaltung neue Produktivitätspotentiale freisetzen, die für die Effizienz freier Software verantwortlich sind und die von kapitalistischen Unternehmen nicht im selben Maße erreicht werden können. Dies – zusammengenommen mit der neuen Technologie, also der einfachen Reproduzierbarkeit und Verbreitung von Wissen – stelle den Kapitalismus grundsätzlich in Frage: „Ist diese Technik an sich schon revolutionär genug, ... so hat die digitale Kopie in Verbindung mit Freier Software und deren Selbstentfaltung erst wirklich systemsprengendes Potential“ (Merten 2001).

Dieses systemsprengende Potential wird vor allem in einer weiteren Eigenschaft Freier Software gesehen: Freie Software befinde sich außerhalb der „Verwertungs-Maschine“ des Kapitalismus: zwar sei das Kapital auch daran interessiert, mit Freier Software Geld zu verdienen und tut es auch, stosse dabei jedoch an Grenzen: „Sie [die GNU Gen-

eral Public License, die Verf.] stellt sicher, dass Software dauerhaft frei bleibt oder ökonomisch formuliert: sie entzieht Software dauerhaft der Verwertung“ (Meretz 2001). In diesem Zusammenhang wird auch auf den „Community-Geist“ der Freien Software-Bewegung verwiesen, der eine Verwertung Freier Software zusätzlich erschwere. So habe die Firma Corel eine GNU/Linux-Distribution vermarkten wollen, ohne jedoch den Quellcode offenzulegen und sei damit gescheitert: „Sehr schnell haben sie (Corel, die Verf.) eingesehen, dass sie sich besser nicht mit der Freien-Software-Bewegung anlegen, die bei Bekanntwerden dieser Fakten empört aufgeschrien hat“ (Merten 2001).

Die Rede von der „Keimform“ impliziert, dass sich aus der Freien Software etwas entwickelt: „Die Freie Software da herausgeholt zu haben [aus dem Verwertungszusammenhang, die Verf.], war eine historische Tat. Jetzt geht es darum, sie draußen zu behalten, und nach und nach weitere Bereiche der kybernetischen Maschine abzutrotzen“ (Meretz 2001). Anscheinend soll Freie Software so eine Art nicht-kapitalistischer Brückenkopf innerhalb des kapitalistischen Terrains sein, von dem ausgehend dann immer neue Gebiete erobert, d. h. der Verwertung entzogen werden, bis für das Kapital schließlich nichts mehr übrig bleibt. Das Ergebnis dieses Prozesses wäre eine neue Gesellschaft, die „GPLGesellschaft“ (Merten in Anlehnung an die General Public License): „Die Freie Software mit ihren Prinzipien jenseits der Verwertung, die das Wort von der Informationsgesellschaft auf den Begriff bringt, scheint die lange gesuchte Keimform zu sein, die eine Vergesellschaftung auf dem Stand der Produktivkraftentwicklung aber jenseits der Tausch- und Arbeitsgesellschaft erstmals aufscheinen lässt“ (Merten 2001).

Zwar sei heute nicht klar, wie so eine „GPLGesellschaft“ genau aussehen würde, es könne allerdings auf Basis der gemachten Erfahrungen mit Freier Software erörtert werden, auf welchen Grundlagen eine solche Gesellschaft beruhen müsste. Diese seien charakterisiert von frei zur Verfügung stehenden Gütern und Wissen. Es wird genommen, was gebraucht wird und nicht gegen Geld getauscht. Die Produktionsmit-

tel müssten „Selbstentfaltung auf breiter Basis“ ermöglichen, es müsse Spass machen, an diesen Produktionsmitteln tätig zu sein. Es gäbe keine Arbeit mehr im herkömmlichen Sinne, es würde nicht mehr für einen Markt produziert werden, sondern aus „konkreten, menschenbezogenen Gründen“ (vgl. Merten 2001). Kurz: es gäbe kein Geld, keine Ware, keine (Lohn)Arbeit, keinen Tausch und die wichtigste Produktivkraft der Menschheit wäre die menschliche Selbstentfaltung.

Wie sich aus der Freien Software als Keimform die „GPL-Gesellschaft“ entwickelt, bleibt uneindeutig. „Keine neue Gesellschaft löst die alte ohne Widerstand ab. Zunächst entwickeln sich Keime des Neuen in den Nischen des Alten. Schließlich wird das Neue so mächtig, dass die Verwalter des Alten Konzessionen machen müssen und das Neue gleichzeitig bekämpfen und verhindern wollen. Das Neue wird sich dann durchsetzen, wenn es effektiv besser ist als das Alte. Dabei ist es klüger, nicht auf dem ureigenen Terrain des Alten zu kämpfen, sondern die Spielregeln zu ändern und sich auf neuem Terrain zu behaupten“ (Meretz 2000). Einerseits klingt hier ein gewisser technischer Determinismus an, andererseits ist aber auch von Kämpfen die Rede. Faszinierend für die „Keimform-Theoretiker“ scheint der Gedanke zu sein, dass man eine grundlegende gesellschaftliche Veränderung erreichen kann, nicht in dem man den Kampf mit dem Alten direkt aufnimmt, sondern indem man das Neue bereits praktiziert. Dabei hat man dann vielleicht mit einigen Widerständen zu tun, das Terrain des Neuen, so die Überzeugung wird sich aber allein schon deshalb ausdehnen, weil es „effektiv besser“ als das Alte ist – ein Gedanke, bei dem unterstellt ist, das Alte und Neues immerhin so nah sind, dass sie die Maßstäbe (wie zum Beispiel für Effektivität) noch teilen!

Entgegnungen

Freie Software wird zum Hoffnungsträger für eine irgendwie „bessere“ Welt. Wie kommt es, dass der Freien Software eine solch wichtige Vorerreiterrolle zuerkannt wird? Erst mal ist Freie Software nichts als ein Produkt mit spezifischen Anwendungsprofilen zur Ausführung und Regulation von Ar-

beitsprozessen (oder Spielen). Freie Software ist als fertiges Produkt, in seinen technischen Eigenschaften von proprietärer Software in nichts zu unterscheiden. Ob sie tatsächlich in einiger Hinsicht besser funktioniert als proprietäre, ist umstritten (kritisch zu den Vorteilen von Linux: Winzerling 2002). Was als entscheidender Vorteil herausgestellt wird, dass es mir als Benutzer möglich ist, die Software zu verändern (weil der Quellcode aufgrund der GPL offen und nicht wie bei proprietärer Software geschützt ist), hat jedoch nur für die kleine Minderheit von Nutzern Bedeutung, die sowohl über die technischen Fähigkeiten als auch über die Zeit für solche Eingriffe verfügen. Die Masse der Nutzer geht mit Freier Software auch nicht anders um als mit proprietärer.

Entzieht sich Freie Software der Verwertung?

Dass sich Freie Software der Verwertung entziehen würde, ist ein oft gehörtes Argument bei jenen, die der Freien Software systemsprengendes Potential beimessen: die Tatsache, dass die GPL den privaten Eigentumsanspruch verhindert, so das Argument, führe zu einem „Überfluss“ an Freier Software, sie ist jedermann frei zugänglich und verunmögliche daher die künstliche Verknappung als Voraussetzung für kapitalistische Verwertung – etwas, was alle kostenlos haben können, ist natürlich schwer verkäuflich. In dieser Wahrnehmung wird offensichtlich „Verkauf“ mit „Verwertung“ gleichgesetzt. Nun ist aber mit den Schwierigkeiten beim Verkauf nur eine Seite des Verwertungszusammenhangs „verwundet“ und zwar jene, die in der Zirkulation stattfindet. In der Produktionssphäre kann Freie Software aber ohne weiteres eingesetzt und zum Bestandteil des kapitalistischen Verwertungsprozesses werden. Dies ist in zweifacher Hinsicht möglich. Zum einen kann Freie Software als kostenloses Produktionsmittel genutzt werden, was im Vergleich zur Verwendung proprietärer Software, die bezahlt werden muss, die Profitrate erhöht. Zum anderen kann Freie Software aus dem Netz gezogen und unter Zusatz von zusätzlicher Arbeit, wie Support oder der Erstellung

von Handbüchern, verkauft werden. Verwertet hat sich dann das vorgeschossene Kapital für die Arbeitskraft und die Produktionsmittel für Handbücher oder/und CDROMs. Die Freie Software hat sich zwar nicht verwertet, weil kein Kapital dafür aufgewendet wurde, sie bildete in diesem Fall aber die Grundlage dafür, dass ein Verwertungsprozeß überhaupt in Gang kam. Freie Software wird bereits auf beide Weisen seit längerem verwendet, Tendenz steigend. [4] Warum gerade Computerproduzenten verstärktes Interesse an Freier Software, insbesondere an Linux haben, macht Werner Winzerling (2002) deutlich: da Microsoft ein Quasi-Monopol für PC-Betriebssysteme besitzt, sei Linux ein willkommenes Gegengewicht, denn auch die Hardware-Produzenten können durch einen Softwaremonopolisten geschädigt werden. In diesem Fall wäre das Interesse an Linux weder in seinen besseren Produkteigenschaften noch in seinem „freien“ Charakter begründet, sondern schlicht und einfach darin, dass es überhaupt eine Alternative zu Windows darstellt.

Freie Software kann als kostenloser Rohstoff angeeignet werden, wie Luft oder Sonne – mit dem einzigen Unterschied, dass menschliche Arbeitskraft darin steckt. Somit kann man folgern: die völlig kostenlose Aneignung fremder Arbeit (nicht einmal Lohn ist zu zahlen) dient als Mittel für einen ganz normal kapitalistischen Verwertungsvorgang. Hier dreht sich der positive Bezug auf Freie Software gewissermaßen um: die unbezahlte Aneignung von fremder Arbeit wird von der GPL legitimiert („allen frei zugänglich“). Der „Überfluss“ an Freier Software stellt für die Kapitalverwertung überhaupt kein Problem da. Dass sich Freie Software der Kapitalverwertung prinzipiell entziehen würde, ist eine Illusion.

Keimform

Wenn wir die Debatten zur „Keimform“, wie sie vor allem im Umkreis von www.oekonux.de geführt werden, richtig verstanden haben, ist mit „Keimform“ ein neues Prinzip gemeint, das zum einen mit dem bestehenden System prinzipiell unverträglich ist und das zum anderen zum Ausgangspunkt einer Unterminierung und schließlich einer Überwindung des alten Systems

werden kann. Dass Freie Software zwar einerseits eine „Anomalie“ für das Privateigentumsparadigma darstellt, dass sie aber mit dem alten System – der Kapitalverwertung – keineswegs unverträglich ist, haben wir oben zu zeigen versucht. Wie steht es aber nun mit der Ausbreitung der „Anomalie“? Als Ansatz dafür, wie man der „kybernetischen Maschine“ weitere Bereiche „abtrotzen“ könne, wird auf freie Projekte im Internet verwiesen, wie z. B. freie Literatur, freie Musik, eine freie Enzyklopädie, usw. Interessant wird es dann aber beim Hinweis auf „Freie materielle Güter“, dort gäbe es auch schon Projekte, die versuchen, die Prinzipien Freier Software auf die Produktion materieller Güter umzusetzen: „Zunächst scheint dies eine unüberwindliche Hürde, da materielle Güter nicht den Bedingungen der digitalen Kopie unterliegen“ (Merten a. a. O.). Dennoch seien einige interessante Entwicklungen zu beobachten. Als Beispiel werden Oscar, die Entwicklung eines Autos, oder die Entwicklung von Entwurfsplänen für elektronische Schaltungen genannt. In diesem Zusammenhang wird dann aber wieder darauf verwiesen, dass zur Realisierung dieser Ideen kommerzielle Firmen den Vorteil hätten, Entwicklungskosten zu sparen: „So gibt es inzwischen mehrere Projekte, die sich mit dem Design materieller Güter befassen. Sie entwerfen dabei ein Gut, das dann von kommerziellen Firmen hergestellt werden kann. Der Vorteil für eine Herstellerfirma liegt darin, dass sie die Kosten für eine Produktentwicklung nicht selbst aufbringen muss“ (Merten 2001). Zum einen wirkt es befremdlich, dass ein Autor, der ansonsten den systemsprengenden Charakter Freier Software betont, nun die Kostenersparnis eines kapitalistischen Unternehmens als „Vorteil“ der Freien Software anpreist. Zum anderen erscheint uns die von Merten als nur „zunächst“ unüberwindlich bezeichnete Hürde vielmehr konstant unüberwindlich zu sein.

Dass Freie Software auf (relativ) breiter Basis produziert werden kann, hat zur Voraussetzung, dass ihre Produktionsmittel – PC und Netzzugang – in den entwickelten kapitalistischen Ländern billig zu haben sind (für die Mehrheit der Menschen, vor allem in den Ländern der „Dritten Welt“, sieht dies erheblich anders aus). Dass die Produktionsmittel

für Freie Software so billig sind, liegt letzten Endes daran, dass es sich hier um „Informationsprodukte“ handelt: das eigentliche Produkt ist die Information (das Programm, oder auch der Plan eines Autos), materiell ist lediglich der Träger der Information. Die Bearbeitung, Speicherung und das Kopieren von Information sind aber relativ einfach, mit wenig Material- und Arbeitsaufwand durchführbar und dadurch billig geworden. Ganz anders sieht es aber bei materiellen Produkten aus. Ein Auto zu bauen erfordert erheblich „mehr“ Aufwand an Produktionsmitteln (und damit auch an Kosten), als ein Software-Tool zu programmieren: ein PC steht auf vielen Schreibtischen, eine Montagehalle mitsamt den entsprechenden Maschinen kann sich nur eine Autofirma leisten. Insofern findet die „Keimform“ an der Welt der kostenaufwendigen und arbeitsintensiven materiellen Produkte ihre Schranke. Von einer Unterminierung kapitalistischer Verhältnisse ist auch in dieser Hinsicht nichts zu sehen.

Selbstentfaltung

Als Besonderheit der Produktion Freier Software wird schließlich noch geltend gemacht, dass hier Menschen kooperieren, deren Tätigkeit durch „Selbstentfaltung“ charakterisiert sei: es ist nicht die Orientierung am Tauschwert der Produkte oder am Lohn, sondern das inhaltliche Interesse am produzierten Gebrauchswert und der Spaß an der Kooperation mit anderen, welche die einzelnen motivieren, ihre Zeit in die Produktion Freier Software zu stecken.

In der Tat ist es beeindruckend, wie es dabei gelingt, dass Menschen weltweit zusammenarbeiten und komplexe Produkte hervorbringen, nicht nur ohne die Motivation des Tauschwertes, sondern auch unter weitgehendem Verzicht auf eine hierarchische Leitungsstruktur. Einer grundsätzlichen Alternative zum Kapitalismus, einer Gesellschaft also, die ohne Geld, Tausch und staatlichen Zwangscharakter der Reproduktion auskommt, einer Gesellschaft, die man als „kommunistisch“ bezeichnen kann (und die mit den in Osteuropa untergegangenen „kommunistischen Staaten“ offensichtlich nichts gemein hat), wird gerne vorgeworfen, sie könne es nicht geben, denn „der Mensch“ sei eben gar nicht so: ohne

äußeren Druck einerseits und materiellen Anreiz andererseits laufe gar nichts und das Ganze müsse außerdem noch durch fähige Leitungspersonen an der Spitze gesteuert werden. Dazu stellt die Produktion freier Software tatsächlich ein Gegenbeispiel dar. Sie macht deutlich, dass selbst unter den Bedingungen des Kapitalismus eine andere Form der Produktion möglich ist – und zwar nicht nur in dem beschränkten Rahmen eines kleinen Projektes, das überschaubar ist und bei dem sich alle kennen, sondern innerhalb eines weltweiten Verbundes. Insofern ist die Produktion Freier Software ein wichtiges Beispiel für die Möglichkeit anderer Kooperationsformen – aber auch nicht mehr. Weder kann sich dieses Beispiel dem kapitalistischen Zugriff entziehen, noch stellt es eine „Keimform“ dar.

Auch dieses Beispiel ist in den kapitalistischen Kontext integriert. Dies gilt nicht nur für ihre Produkte, die keineswegs jenseits des kapitalistischen Verwertungszusammenhangs stehen, dies gilt auch für die auf „Selbstentfaltung“ beruhenden Produktionsbedingungen. Dass Menschen im Kapitalismus in dieser Weise kooperieren können, hat zur Voraussetzung, dass einerseits ihr Lebensunterhalt gesichert ist (entweder indem sie Lohnarbeit leisten oder z. B. als Studenten staatlich alimentiert werden) und dass ihnen die Sicherung ihres Lebensunterhaltes andererseits genügend Zeit lässt, sich mit Freier Software zu beschäftigen. Betrachtet man die kapitalistischen Verhältnisse weltweit, dann gehören diejenigen, die Freie Software entwickeln, zu einer kleinen, privilegierten Gruppe innerhalb der entwickelten kapitalistischen Länder. Diese Privilegierung lässt sich unter kapitalistischen Bedingungen wohl kaum verallgemeinern. Aber auch für diejenigen, die diese Privilegierung heute genießen können, besteht immer die Gefahr, dass sich ihre Situation aufgrund von Krisenprozessen ändert: dass sie arbeitslos werden oder die Arbeitsintensität steigt, dass Ausbildungsförderung gestrichen oder der Druck in den Ausbildungsinstitutionen erhöht wird. Insofern liegen auch die ProduzentInnen Freier Software lediglich an der (im Moment etwas längeren) Leine des Kapitals.

Aber selbst die angesprochene „Selbstentfaltung“ entzieht sich nicht gänzlich der Verwertung.“ Selbstentfaltung“ ist eine Produktivkraft, die auch das Kapital seit einiger Zeit für sich entdeckt. So hat Norbert Bensel, verantwortlich für Human Resources der DaimlerChrysler Services AG, jüngst bei einem Vortrag auf der Konferenz „Gut zu Wissen“ der Heinrich-Böll-Siftung neue Arbeitskonzepte vorgestellt, die in ihrem nicht nur sprachlichen Habitus sehr den Selbstentfaltungsaspekten, die der Freien Software zugeschrieben werden, ähneln. Er verwies auf das Modell OpenSource als Vorbild und beschrieb das neue Arbeitsmodell mit folgenden Stichworten: „Spas haben“ (statt Geld verdienen, als Motivation...), „Freiwillige motivieren“, „Anerkennung für cool code“, „Kunden zu Mitarbeitern machen“, „Bedürfnis der Mitarbeiter nach Entfaltung“ usw. Im Abstract zu seinem Vortrag heisst es unter anderem: „... gängige Strukturmodelle mit einer festen Befehlshierarchien von oben nach unten spiegeln die betriebliche Realität nicht mehr wieder...“ (Bensel 2001).

Auch den Anhängern der „Keimform-Theorie“ ist dieser Sachverhalt bekannt. So schreibt Stefan Meretz: „... die Sachwalter des Kapitals als Exekutoren der Wertverwertungsmaschine haben erkannt, dass der Mensch selbst die letzte Ressource ist, die noch qualitativ unentfaltete Potenzen der Produktivkraftentwicklung birgt. In seiner maßlosen Tendenz, alles dem Verwertungsmechanismus einzuverleiben, versucht das Kapital auch diese letzte Ressource auszuschöpfen“ (Meretz 2000). Allerdings stoße das Kapital dabei an Grenzen: eine wirklich freie Selbstentfaltung sei nicht möglich, da in einem kapitalistischen Unternehmen letzten Endes doch die Verwertungsaspekte und die Konkurrenz der Mitarbeiter untereinander dominieren würden. Daher sei es dem Kapital gar nicht möglich, die in der Selbstentfaltung steckende Produktivkraft wirklich auszuschöpfen.

Wie die Idee der Freien Software, bzw. die damit zusammenhängende „Selbstentfaltung“ von der real existierenden Welt vereinnahmt wird, ist aber unserer Ansicht nach nicht wegzuwischen mit dem Argument, dass Diskussionen über die Kompatibilität von Freier Software

und Kapitalismus an der Sache vorbeigehen und dass man vorwiegend darüber reden müsse, wie überhaupt „die Arbeit beschaffen sein muss, damit sich in ihr der Mensch als Subjekt voll entfalten kann“ (Meretz 2000). Dies ist eine Form von Utopismus: es wird unter Absehung der realen Entwicklungen ein Gegenbild entworfen, ein nettes Märchen erzählt, darüber wie es aussehen würde, wenn das Rotkäppchen nicht zum Wolfe rennen würde. Derweil aber ist der Wolf gerade dabei, das Rotkäppchen zu fressen.

Unsere Gegenthese lautet: Es gibt keinen unüberwindlichen Gegensatz zwischen Kapitalismus und Freier Software. Im Gegenteil: die Digitalisierung wie auch die Entstehungsbedingungen Freier Software scheinen eher einer Modernisierung Vorschub zu leisten, bzw. sind selbst Antrieb der Modernisierung der gegenwärtigen Produktionsweise, die sich in flexibleren, dezentralisierten, globalisierten und atomisierten Arbeits- und Produktionsbedingungen niederschlägt, die der einzelnen Arbeitskraft mehr Verantwortung für das Produkt überträgt: „Ein global erfolgreiches Unternehmen, das sich ausschließlich auf den Grundlagen einheimischer Werte aufbaut, verschliesst sich einem Reichtum an Talenten, Ideen und unterschiedlichen Sichtweisen. ‚Managing Diversity‘ wird damit zur zweiten großen Herausforderung an die Führung eines Unternehmens“ (Bensel 2001).

Tatsächlich zu beobachten ist ein Wandel der Eigentumsverhältnisse. Mittels der Digitalisierung und der Computerisierung der Individuen befinden sich einige der entscheidenden Produktionsmittel nicht mehr ausschließlich im Privateigentum der Unternehmen, sondern auch im Privateigentum der Arbeitskräfte. Dies führt aber nicht zur Aufhebung des Kapitalismus, sondern zu einer tendenziell anderen Qualität des Kapitalismus: es ist nicht mehr allein das Produkt, das gekauft oder verkauft wird und dessen inkarniertes Wissen geheim gehalten werden muss, sondern es ist der ganze Mensch, den das Unternehmen benötigt. Das Wissen, in digitalisierter Form, kann zunehmend schwerer eingegrenzt, bzw. künstlich verknappt werden, als Rohstoff kann es frei zugänglich bleiben: „Die Bedeutung des

Mitarbeiters, der ‚Human Resources‘, hat sich für Unternehmen entscheidend gewandelt: Information und Wissen sind der Motor der modernen Dienstleistungsgesellschaft. Dabei ist jedoch der effiziente Zugriff auf Information letztlich nicht der entscheidende Wettbewerbsfaktor. Es sind die Menschen, die Mitarbeiter, die mit ihrem Wissen aus der Information neues Wissen erzeugen“ (ebd.). Die Unternehmen reagieren auf diese Entwicklung entsprechend: „Generell ist ein Trend weg vom geregelten Arbeitstag mit Anwesenheitspflicht hin zur Vertrauensarbeitszeit zu verzeichnen. Entsprechend halten Arbeitszeitkonten, Langzeit- und Lebensarbeitszeitmodelle Einzug in Unternehmen. Die Leistung des Einzelnen wird zunehmend nicht mehr an der Anwesenheitszeit im Unternehmen, sondern an den erreichten Zielen und der Qualität der Ergebnisse gemessen. Das Angebot von Telearbeitsplätzen oder die Vereinbarung von Teilzeitarbeitsverträgen sind heute in vielen Unternehmen schon Realität“ (ebd.).

Die Unternehmen sind also schon längst dabei, die Potentiale der „Selbstentfaltung“ auszunutzen. Wenn es dabei Grenzen gibt, könnte dies Freie Software zusätzlich attraktiv machen: vielleicht dauert es nicht mehr lange, bis Unternehmen die Produktion Freier Software durch Stipendien oder ähnliches fördern: ganz selbstbestimmt könnte dann Freie Software produziert werden, was lediglich den bisherigen Software-Monopolisten wie etwa Microsoft weh tun würde, der Unternehmenssektor als Ganzer könnte aber gerade von der GPL profitieren. Die Mehrzahl derjenigen, die Freie Software entwickeln, dürfte damit kein Problem haben: es ist ja nicht nur so, dass ihre Arbeitsmotivation aus dem Interesse am Produkt stammt, bei vielen Mitgliedern der Freien-Software-Gemeinde beschränkt sich das politische Interesse auch auf die möglichst ungestörte Produktion, mit der Kompatibilität von Freier Software und Kapitalismus haben sie kein Problem.

Vor diesem Hintergrund hat dann auch die GPL nichts Revolutionäres mehr und der Satz von Stefan Meretz: „Linux als Entwicklungsmodell nimmt einiges

der neuen Gesellschaft vorweg“ könnte, überspitzt formuliert, dann auch anders gelesen werden: Die „neue Gesellschaft“ ist der modernisierte Kapitalismus.

[1] In den „Enclosures“ (Einhegungen) des früheren Gemeindelandes, die einerseits Produktionsmittel zu Privateigentum machten und andererseits die früheren Nutzer dieser Produktionsmittel zu „freien“ Arbeitern, denen nichts anderes übrig blieb, als ihre Arbeitskraft zu verkaufen, sah Marx den zentralen Prozeß der „ursprünglichen Akkumulation“ in England, der Geburtsstätte des modernen Kapitalismus

[2] Der Quellcode ist in einer menschenlesbaren Sprache geschrieben. Dieser Code muss erst von einem „Compiler“ übersetzt werden, damit der Computer ihn versteht. Der Quellcode ist nach der Übersetzung für den Ablauf des Programms nicht mehr notwendig, man kann ihn weglassen. Eine Rückübersetzung von einem maschinenlesbaren in einen menschenlesbaren Code ist dann aber wieder nahezu unmöglich, man braucht den Quellcode, wenn man etwas am Programm ändern will.

[3] Die Lizenz gewährt die freie Benutzung des Programms, Kopien des Programms zu erstellen und zu verbreiten, das Programm zu modifizieren und modifizierte Versionen zu verteilen. Die GNU GPL schreibt vor, dass der Quellcode frei, jederzeit verfügbar sein und bleiben muss, dass die Lizenz eines GPL-Programms nicht geändert werden darf und dass ein GPL-Programm nicht Teil nicht-freier Software werden darf. Sie verbietet es, GPL-Programme in proprietäre Software zu überführen.

[4] Anwender von freier Software sind z. B. Lehmanns Buchhandlung, die tageszeitung, Babcock- BSH, Individual Network. Mit Freier Software verbundene Dienstleistungen verkaufen Firmen wie SuSE, Lunetix, innominate, New Technologies Management GmbH (vgl. Grassmuck 2000).

*) Die folgende Text ist die gekürzte Fassung eines Vortrags, den wir bei der ersten ÖkonouX Konferenz in Dortmund 28. -30. April 2001 gehalten haben.

Literatur

- Bensel, Norbert: *Arbeitszeit, Weiterbildung, Lebenszeit – neue Konzepte*, in: Plenarbeiträge vom Kongress „Gut zu Wissen“, Links zur Wissensgesellschaft, 4. -6. Mai 2001, Berlin, S. 10ff.
- Grassmuck, Volker: *Freie Software, Geschichten, Dynamiken und gesellschaftliche Bezüge*. September, 2000, <http://mikro.org/Events/OS/text/freesw.pdf>.
- Kuhn, Thomas (1962): *Die Struktur wissenschaftlicher Revolutionen*, Frankfurt/M 1973.
- Meretz, Stefan (2000): *LINUX & CO. Freie Software – Ideen für eine andere Gesellschaft*. Quelle: <http://www.kritischeinformatik.de/index.htm?fsrevo.htm>.
- Meretz, Stefan (2001): *GNU/Linux ist nichts wert – und das ist gut so!*, <http://www.oekonouX.de/texte/werthos/index.html>.
- Merten, Stefan (2001): *Freie Software für eine Freie Gesellschaft. Bringen Gnu/Linux und Co uns einer neuen Gesellschaft näher?* <http://www.oekonouX.de/texte/neuegesellschaft/index.html>.
- Nuss, Sabine (2002): *Download ist Diebstahl? Eigentum in einer digitalen Welt*, in: PROKLA 126 (März 2002).
- Winzerling, Werner (2002): *Linux und Freie Software: Eine Entmystifizierung*, in PROKLA 126 (März 2002).

Michael Heinrich: Michael Heinrich ist Politikwissenschaftler in Berlin und Autor des Buches „Die Wissenschaft vom Wert“, 2. erw. Auflage, Münster, Verlag Westfälisches Dampfboot.

Sabine Nuss: Leiterin der Politischen Kommunikation bei der Rosa-Luxemburg-Stiftung und Co-Geschäftsführerin des Karl Dietz Verlags Berlin.

Lizenz dieses Beitrags

LFK

Lizenz Freie Kunst