Dieser Artikel ist Teil des

Open Source Jahrbuch 2005

erhältlich unter http://www.opensourcejahrbuch.de.

Das Open Source Jahrbuch 2005 enthält neben vielen weiteren interessanten Artikeln ein Glossar und ein Stichwortverzeichnis.

Freie Software und Freie Gesellschaft

Stefan Merten und Stefan Meretz

Der weltweite Erfolg freier Software hat die Frage aufgeworfen, ob ihre Prinzipien verallgemeinerbar und auf die gesamte Gesellschaft übertragbar sind. 1999 gründete sich das Oekonux-Projekt rund um diese Frage. Oekonux steht für „Oekonomie und GNU/Linux". Zahlreiche Analysen aus unterschiedlichen Richtungen haben zu Thesen geführt, die viele für bahnbrechend halten und an denen sich viele reiben – innerhalb und außerhalb des Projektes. Der erste Teil des Beitrages beschäftigt sich mit den Eigenschaften freier Software und dessen Prinzipien. So wird zuerst die Produktionsweise von freier Software betrachtet, der Unterschied zwischen einfach und doppelt freier Software gezeigt und anschließend erklärt, warum es sich bei freier Software um eine neue Qualität der Produktivkraftentwicklung handelt. Diese Betrachtungen sollen im letzten Teil dazu dienen, in freier Software eine Keimform für eine neue Gesellschaft zu sehen. Diese Betrachtung wird mit dem Bild der „GPL-Gesellschaft" abgeschlossen.

1. Einleitung

Freie Software hinsichtlich eines möglichen Ausgangspunkts einer neuen, fundamental veränderten Gesellschaftsformation zu untersuchen, hat sich das Projekt Oekonux zur Aufgabe gemacht. Das ganz überwiegend virtuelle Projekt gruppiert sich um mehrere Mailing-Listen und einige Websites. Im nicht-virtuellen Raum trat das Projekt mit bisher drei internationalen Konferenzen an die Öffentlichkeit.[1] Der vorliegende Artikel, der diese zentralen Thesen beleuchtet, wurde in einem offenen Prozess entwickelt,[2] der auch jetzt noch für Beiträge offen ist. Der Artikel steht nicht nur zum Lesen zur Verfügung, sondern kann auch direkt online im Browser Absatz für Absatz kommentiert werden.

1 Informationen zum Projekt und den bisherigen Konferenzen befinden sich unter http://www.oekonux.de/projekt/index.html.

2 Der Beitrag befindet sich online unter unter http://www.opentheory.org/ox_osjahrbuch_2005/.

2. Produktionsweise freier Software

Die Produktionsweise freier Software unterscheidet sich grundsätzlich von der proprietärer Software. Dies betrifft weniger die technischen Verfahren, sondern vor allem die individuelle Motivation und die soziale Organisation. Diese Produktionsweise ist gekennzeichnet durch *Wertfreiheit*, *Selbstorganisation*, *Globalität* und *Selbstentfaltung*. Diese vier zentralen Begriffe sollen einleitend kurz umrissen und im restlichen Text aus verschiedenen Perspektiven weiter erläutert werden (Vgl. Merten 2000, Meretz 2000*b*, 2004).

Die Entwicklung von Software ist mit Aufwand verbunden. Bei freier Software wird dieser Aufwand in der Regel jedoch nicht (monetär) entlohnt. Wie auf vielen anderen Gebieten menschlichen Lebens strengen sich die Menschen hier aus anderen Gründen an, als Geld dafür zu erhalten. Das Resultat dieser Tätigkeit ist deswegen ökonomisch *wertfrei* und unterscheidet sich damit wesentlich von der wertbasierten Arbeit, die auf die Erreichung von Lohn oder Profit abzielt (Vgl. Meretz 2000*a*).

Die sozialen Organisationsformen freier Software sind so verschieden wie die Projekte selbst. Niemand gibt von außen vor, wie etwas zu sein hat. Jedes Projekt organisiert sich selbst und findet die Form, die ihm gemäß ist, oft einfach durch Ausprobieren.

Die *globale Vernetzung* istdas Resultat der Möglichkeiten des Internets. Jedes noch so kleine Projekt, das sich einer der vielen frei zugänglichen Projekt-Infrastrukturen bedient (SourceForge, Savannah etc.) oder selbst eine betreibt, ist weltweit verfügbar. Menschen, die sich noch nie gesehen haben und vielleicht auch niemals sehen werden, können so zusammen etwas Nützliches erschaffen. Ohne das Internet mit seinen vielfältigen Diensten wie E-Mail, FTP oder WWW wäre freie Software in ihrer heutigen, entfalteten Form nicht denkbar.

3. Hauptantrieb Selbstentfaltung

Selbstentfaltung ist ein zentraler Begriff für das Verständnis freier Software. Selbstentfaltung meint nicht einfach „Spaß haben“ und besitzt eine individuelle und eine gesellschaftliche Dimension. Individuell meint Selbstentfaltung das persönliche Entfalten der eigenen Möglichkeiten, mithin das Entwickeln der eigenen Persönlichkeit. Eine so verstandene Entfaltung der Persönlichkeit hat verschiedene Formen der Entäußerung: produktive, reproduktive, technische, kulturelle, kommunikative, konsumtive etc. Diese können für andere nützlich sein (Vgl. Himanen 2001).

Die gesellschaftliche Dimension der Selbstentfaltung betrifft die Abhängigkeit der eigenen Entfaltung von der Entfaltung der anderen. Ich kann mich nur entfalten, wenn die anderen es auch tun. Die anderen – potenziell alle anderen – sind meine Entfaltungsbedingung, wie ich umgekehrt Entfaltungsbedingung für die anderen bin. Es entsteht eine positive Rückkopplung: Mein Bestreben richtet sich darauf, dass die anderen sich entfalten können, damit ich mich entfalten kann. Würde ich mich nur darauf konzentrieren, was ich zu tun wünsche und die anderen ignorieren oder gar ausgrenzen, dann würde ich mir selbst schaden.

Diese Dynamik können wir, mehr oder weniger ausgeprägt, bei freier Software beobachten. Die positive Rückkopplung kommt zustande, weil und wenn es keine dritten, entfremdeten Gründe gibt, tätig zu werden. Da freie Software nicht für den Verkauf produziert wird, gibt es keine entfremdeten Gründe, sondern nur jeweils meine Gründe, freie Software zu entwickeln oder zu unterstützen. Proprietäre Software hingegen wird für einen dritten, fremden, der Software äußerlichen Zweck entwickelt.[3]

Im Projekt Oekonux wurde diese Analyse zu dem Satz verdichtet: „Die Selbstentfaltung des Einzelnen ist die Bedingung für die Entfaltung Aller – und umgekehrt". Besonders deutlich wird die Unterscheidung, wenn man den gleichen Satz für die entfremdete Warenproduktion formuliert. Dort gilt: Die Entwicklung des Einzelnen ist möglich auf Kosten der Entwicklung der anderen – und umgekehrt.

Selbstentfaltung darf nicht mit Selbstverwirklichung verwechselt werden. Während Selbstentfaltung die anderen als Bedingung für die eigene Verwirklichung versteht, blendet Selbstverwirklichung die gesellschaftliche Dimension aus. Selbstverwirklichung ist statisch und begrenzt, sie geht von einer Anlage aus, die verwirklicht werden will und endet mit der Verwirklichung. Selbstentfaltung hingegen ist dynamisch. Jede erreichte Entfaltung ist wiederum nur Bedingung und Möglichkeit neuer Formen der Entfaltung. Eine Gesellschaft der Selbstentfaltung wäre eine reiche Gesellschaft (Gorz 2004).

4. Einfach und doppelt freie Software

In der freien Software-Bewegung wird von verschiedenen Seiten immer wieder betont, dass freie Software Bestandteil von Geschäftsmodellen sein kann. Es soll möglich sein, mit freier Software Geld zu verdienen.

Nun ist klar, dass Geschäftsmodelle, die mit einer Verknappung eines fertigen Produkts operieren, bei freier Software nicht funktionieren können. Verknappung von Informationsgütern ist unter den Bedingungen der weltweiten digitalen Kopie allgemein nur zu erreichen, wenn den NutzerInnen das Recht genommen wird, selbst das Informationsgut weiterzugeben. Die Grundrechte freier Software – uneingeschränkte Einsatzmöglichkeiten, Einsicht in und Möglichkeit zur Anpassung der Quellen, unbeschränkte Weitergabe originaler oder veränderter Versionen – erlauben diese Verknappung jedoch nicht.

Es ist zwar nicht möglich, fertige Produkte direkt zu verkaufen, dennoch gibt es verschiedene Dienstleistungen rund um freie Software, die verkauft werden können: beispielsweise Wartung, Installation oder Zusammenstellen von Distributionen. Diese Geschäftsmodelle arbeiten zwar mit vorhandener, fertiger freier Software, verkauft wird aber letztlich die Dienstleistung. Kaufe ich beispielsweise eine Distribution von SuSE/Novell, so kaufe ich ein Handbuch, CDs und DVDs sowie das Recht auf telefonischen Support für die Installation, das sogar durch einen speziellen, individuellen Code abgesichert ist. Die freie Software, die auf den mitgelieferten Medien

3 Der Markt entscheidet, ob die Software überlebt und wenn sich zu wenige KäuferInnen finden, verschwindet die Software. Das gibt es bei freier Software nicht. Solange sich eine EntwicklerIn für die Software interessiert, gibt es sie – auch, wenn es keine aktuellen NutzerInnen mehr gibt.

enthalten ist, kann ich mir jedoch ganz legal und ohne Bezahlung auch direkt von den SuSE-FTP-Servern downloaden. Tatsächlich bezahlt wird SuSE also für die materiellen Produkte, die sich auf die verteilte freie Software beziehen sowie für Support-Dienstleistung. Grundlage für dieses Geschäftsmodell ist die Dienstleistung der Zusammenstellung und Pflege der Distribution aus dem riesigen Pool vorhandener freier Software. Auch die Anbieter verschiedener Merchandising-Produkte rund um einzelne freie Software-Projekte oder auch Bücher zu freier Software verkaufen nicht freie Software, sondern eben Plüschpinguine.

Ist mit fertiger freier Software selbst kein Geschäftsmodell zu begründen, so gilt das nicht für freie Software, die noch nicht existiert. Es sind durchaus Geschäftsbeziehungen möglich, bei denen freie Software *im Auftrag* erstellt wird. Solche Auftragsarbeiten unterscheiden sich im Falle von Projekten ohne Anbindung zu einer Community von proprietärer Software lediglich durch die Lizenz, unter der das fertige Produkt später steht. Ebenfalls in diese Kategorie fällt die Weiterentwicklung freier Software, die in Firmen für eigene Zwecke durchgeführt wird.

Gegenüber der klassischen freien Software, wie sie ein Richard Stallman oder ein Linus Torvalds entwickelten, gibt es bei den genannten Geschäftsmodellen jedoch einen wichtigen Unterschied. Wie, wohin und wie schnell sich freie Software entwickelt, die ohne externen Auftrag entsteht, liegt allein in der Entscheidung des jeweiligen Projekts. Zu den Freiheiten, die die Lizenzen den NutzerInnen gewähren, tritt in diesen Fällen die Freiheit der EntwicklerInnen, die nicht an Weisungen von Auftraggebern gebunden sind. In Fällen, wo allein die Selbstentfaltung der EntwicklerInnen den Fortgang des Projekts bestimmt, sprechen wir von *doppelt freier Software*.

Demgegenüber sprechen wir von *einfach freier Software*, wenn die EntwicklerInnen in ihren Entscheidungen nicht frei, sondern an einen Auftraggeber gebunden sind. Die EntwicklerInnen entfremden sich in solchen Projekten von ihrem Produkt, da sie auf Grund der Abhängigkeit von der Bezahlung Entscheidungen des Auftraggebers berücksichtigen müssen, die aus ihrer Sicht für das Produkt schädlich sein können. Alle, die schon einmal Software im Auftrag einer Firma hergestellt haben, kennen unzählige Beispiele für von Marketing oder Vertrieb bestimmte Terminpläne, technisch überflüssige Hochglanzfeatures usw. Hier zeigt sich deutlich die Entfremdung vom Produkt, die mit der Erfüllung des dem Produktnutzen äußerlichen Zwecks der Verwertung entsteht.

Betrachten wir freie Software als neue Produktionsweise, so tritt gerade die freie Entscheidung der EntwicklerInnen in den Vordergrund. Nicht getrieben von Marktvorgaben, mithin frei von den Zwängen der Verwertung, können sich die EntwicklerInnen auf die bestmögliche Qualität der Software konzentrieren. Unter anderem ist es möglich, wie jüngst bei GIMP geschehen,[4] sich eine zweijährige Auszeit zu nehmen, in

4 Aus c't 13/2004, S. 80: „Vor-Meilenstein – Open-Source-Bildbearbeitung GIMP: Fit für die Profi-Liga?" Doch die eigentliche Dreijahresleistung der Entwickler fand hinter den Kulissen statt, wie GIMP-Maintainer Michael „Mitch" Natterer gegenüber c't erklärte: „Mit dieser Version wurde die Programmlogik strikt vom Frontend getrennt, kein Quellcode-Stein blieb auf dem anderen. In GIMP 1.2 befanden sich sämtliche Quellcode-Dateien in einem Verzeichnis; niemand wusste genau, was wozu gehört. Kleine Änderungen an einer Stelle konnten anderswo verheerende Auswirkungen haben." Für die Entwickler

der die historisch gewachsene Code-Basis durch ein neues, qualitativ hochwertigeres Fundament ersetzt wird. Anstatt neue Features zu implementieren, wird hier die langfristige Qualität in einer Weise gesichert, wie man sie sich von bekannten proprietären Software-Produkten wünschen würde.

Der qualitative Vorsprung doppelt freier Software ist struktureller Natur und kann auf Grund der in einfach freier Software angelegten Entfremdung von dieser nicht eingeholt werden. Dieser qualitative Vorsprung ist es aber letztlich, der dem Produktivkraftmodell doppelt freier Software den entscheidenden Vorteil vor dem einfach freier oder proprietärer Software gibt. Gäbe es diesen Vorsprung nicht, so hätte freie Software bei Software-NutzerInnen keine Chance gegen proprietäre Software gehabt. Der heute zu beobachtende Erfolg freier Software basiert im Fundament nämlich auf der doppelt freien Software, die teilweise schon vor vielen Jahren geschrieben wurde.

Dennoch treibt auch einfach freie Software insgesamt die Entwicklung in Richtung einer freien Verfügung über freie Produkte voran und in der Praxis mischen sich beide Formen oft. Dies geschieht insbesondere dann, wenn eine doppelt freie Community zu EntwicklerInnen einfach freier Software hinzutritt.

5. Freie Software ist keine Ware

Wollen wir ermessen, inwiefern sich freie Software vom vorherrschenden Wirtschaftsmodell der Marktwirtschaft unterscheidet, so ist es sinnvoll, freie Software mit einem der zentralen Elemente der Marktwirtschaft zu vergleichen: *Der Ware*. Unter Waren verstehen wir in diesem Kontext Güter, die primär zum Zwecke des Verkaufs auf einem Markt produziert werden und sich also von Gütern unterscheiden, die primär aus anderen Gründen produziert werden, z. B. weil sie nützlich sind. Für am Markterfolg orientierte Produkte genügt bei den ProduzentInnen ein *relativer Qualitätsanspruch*, da es lediglich darum geht, in den Augen der avisierten KäuferInnen besser zu sein als die Konkurrenz. Während sich die Qualität von marktorientierten Produkten in dieser Relativität erschöpft und in Monopolsituationen zu deutlich sichtbarer mangelnder Qualität führt, liegt bei einer Produktion, bei der die Nützlichkeit eines Produkts im Vordergrund steht, ein *absoluter Qualitätsanspruch* in der Logik der ganzen Produktionsweise.

Für die Wareneigenschaft ist der Preis der Ware im Übrigen nicht relevant und preislose Waren sind uns spätestens seit den Lockangeboten für Handy-Verträge vertraut.[5]

---

stand außer Frage, dass sie mit dieser Struktur an einem toten Punkt angelangt waren. „Um an wirklich neue Features wie CYMK, 16 Bit oder Ebenengruppen überhaupt denken zu können, musste das Ganze erst einmal auf ein ordentliches Fundament gesetzt werden", konkretisiert Natterer die Probleme des alten GIMP. Heute befinden sich die internen Funktionen von GIMP in 12 klar voneinander getrennten Modulen. Die Oberfläche wurde fast komplett neu geschrieben und ebenfalls in Module aufgeteilt. „Dabei hat fast jede Zeile Quellcode mehr als einmal ihren Platz gewechselt", umreißt Natterer das Mammutprojekt.

5 Ein Sonderfall ist Freeware, die keine freie Software ist. Freie Software zeichnet sich durch freie Verfügung, freie Quellen, freie Änderbarkeit und freie Verteilbarkeit aus. Freeware zeichnet sich dagegen nur durch Kostenfreiheit aus und muss die Punkte, wie sie bei freie Software gefordert sind, nicht erfüllen.

Freie Software hat jedoch in der Regel keinen Preis mehr, sobald sie einmal veröffentlicht ist. Viele, die an die geldbasierte Gesellschaft gewöhnt sind, sind zunächst einmal skeptisch gegenüber dieser Tauschfreiheit. Sie erwarten, dass Güter, für deren Erhalt sie nichts oder unverhältnismäßig wenig geben müssen, entweder Teil der Werbung sind oder sonst einen Pferdefuß haben. Freie Software ist aber weder Werbung noch hat sie sonst einen Pferdefuß. Insbesondere doppelt freie Software ist vielmehr von Anfang bis Ende jenseits des Tauschprinzips angesiedelt. Auch wenn die Teilnahme an einem freien Software-Projekt Geben und Nehmen beinhaltet, so ist der Erhalt von Leistungen jedoch nicht an die Erbringung von Leistungen gekoppelt. Tatsächlich werden die allermeisten NutzerInnen freier Software wenig oder gar nichts zu deren Weiterentwicklung leisten, können sie aber dennoch völlig uneingeschränkt nutzen.

Auf Grund der Konkurrenz sind Betriebsgeheimnisse in der Warenproduktion unerlässlich. Bei freier Software liegen dagegen die Quellen offen vor, sodass es gar keine Geheimnisse geben kann. Alle Interessierten können jederzeit das gesamte Know-how verwenden, das in einer Software enthalten ist.

Gleichzeitig lädt die Offenheit die NutzerInnen ein, die Software zu benutzen und Fehler und Wünsche zu melden, und sie lädt EntwicklerInnen ein, Verbesserungen und Erweiterungen einzubringen. Jeder auch noch so kleine Beitrag bringt alle voran. Freie Software saugt Kreativität und Wissen an. So herrscht Überfluss nicht nur beim Nehmen, sondern auch die Hineingabe ist potenziell unbegrenzt. Freie Software lädt zur *Kooperation* ein, sie funktioniert nach einem *Inklusionsmodell*.

Konkurrenz, also Durchsetzung auf Kosten anderer, wie wir sie zwischen Warenproduzenten erleben, gibt es insbesondere bei doppelt freier Software nicht. Wo es für eine bestimmte Problemstellung mehrere Programme gibt, so beziehen sie sich nicht konkurrenzförmig, also negativ, aufeinander. Entweder existieren die Projekte ohne besondere Beziehung nebeneinander oder es gibt eine mehr oder weniger starke Kooperation zwischen den Projekten. Proprietäre Software muss dagegen nicht nur das Nehmen begrenzen, sondern auch die Hineingabe ist beschränkt, denn nur ausgewählte EntwicklerInnen dürfen in den Quelltext sehen. Unsichere Software ist oft die Folge. Proprietäre Software basiert auf einem *Exklusionsmodell*.

Nun hat es in der Vergangenheit immer wieder Produktionsformen gegeben, die nicht vom Warenmarkt ausgegangen sind. Nicht selten sind Produkte zunächst im Hobbybereich ersonnen worden, und die Wirtschaft hat diese Erfindungen aufgegriffen. In solchen Fällen ist dem Hobbybereich bestenfalls eine Nische geblieben.

Anders bei freier Software. Wurde Software in der Frühzeit der Computer nicht als eigenständige Ware begriffen, so hatte sich in den 80er Jahren des vergangenen Jahrhunderts ein Warenmarkt für Software etabliert. Freie Software, die auf einem anderen Produktivkraftmodell als dem der Warenproduktion basiert, trat nun aus Sicht der NutzerInnen in direkte Konkurrenz zur proprietären Software. Im Gegensatz zu allen früheren Beispielen konnte sich freie Software aber nicht nur eine Nische sichern, sondern wächst im Gegenteil immer weiter und wird nach und nach zu einer ernsten Bedrohung für die proprietäre Software. Das neue Produktivkraftmodell, das wir in freier Software erkennen können, hat das Zeug dazu, die etablierte Warenwirtschaft zu ersetzen.

Freie Software ist nicht zuletzt im *Überfluss* vorhanden. Allein diese Eigenschaft ist ein nachhaltiges Hindernis, freie Software zu einer Ware zu machen. Um den Zusammenhang zwischen Produktion, Konsumption und Gesellschaft übergreifender zu verstehen, lohnt sich eine differenziertere Betrachtung der Begriffe Vorkommen, Begrenztheit und Knappheit.

Mit *Vorkommen* meinen wir, dass ein Gut vorkommt, unabhängig davon, ob wir es brauchen oder nicht. Die gesellschaftliche Dimension ist in diesem Begriff also nicht enthalten. Vorkommen kennt ein absolutes Maß, das z. B. im Begriff des Rohstoffvorkommens gefasst ist. Verleiht man dem Begriff ein zeitliches Maß, so ist er auch auf hergestellte Güter, also Produkte, übertragbar und er bezeichnet die zu einem bestimmten Zeitpunkt existierenden Produkte.

Mit *Begrenztheit* bezeichnen wir das Verhältnis zwischen der Verfügbarkeit eines Gutes und den Bedürfnissen der Menschen, dieses zu erhalten und zu nutzen. Gemessen am Bedarf, kann ein Gut in zu geringer, eben begrenzter Menge zur Verfügung stehen. Solche Begrenzungen können durch gesellschaftliches Handeln abgestellt werden, indem im einfachsten Fall vom begrenzten Gut mehr hergestellt wird. Produktion im allgemeinen Sinne bedeutet immer, gesellschaftlich mit Begrenzungen umzugehen.

Eine besondere Form des Umgangs mit Begrenzungen ist die Warenproduktion. Eine Ware darf nicht frei verfügbar sein, sonst ist sie keine, sie muss knapp sein. *Knappheit* ist eine geschaffene, soziale Form der Warenproduktion. Sie ignoriert wirkliche Begrenzungen und Vorkommen, um daraus die real wirksame Form Knappheit zu machen. Die soziale Form Knappheit produziert die Paradoxie des Mangels im Überfluss. Da abgelöst vom wirklichen Vorkommen, kann sie auch nicht nachhaltig sein.

6. Das Maintainer-Modell

Neben der Wertfreiheit spielt auch die Organisationsform eine große Rolle. Wer länger in der Softwareentwicklung tätig ist, weiß, dass bei Software-Projekten der soziale Prozess bezüglich der Organisation eine wesentliche Rolle spielt. Ist der soziale Prozess schlecht oder gar nicht organisiert, versinkt das interessanteste Projekt im Chaos, und Kreativität und Produktivität der Aktiven werden nachhaltig gestört. Wie ist dieser Prozess in freien Software-Projekten – genauer: in Projekten doppelt freier Software – organisiert?

Der entscheidende Unterschied zwischen der Entwicklung doppelt freier Software und anderer Software-Entwicklung besteht darin, dass alle ProjektteilnehmerInnen ausschließlich auf der Basis von *Freiwilligkeit* am Projekt teilnehmen. Da die Aktiven durch keinerlei entfremdete Anreize, wie zum Beispiel Entlohnung, an das Projekt gebunden sind, können sie das Projekt auch genauso freiwillig wieder verlassen.

So wie die Teilnahme nicht durch äußerliche Aspekte des Projektes bestimmt ist, ist es auch das Projekt insgesamt nicht. Vielmehr kann und muss sich jedes Projekt selbst Ziele setzen, sich *selbst organisieren*. Die Ziele beziehen sich dabei ausschließlich auf das gemeinsame Produkt und dessen Qualität.

Die Bedingungen der Freiwilligkeit der TeilnehmerInnen und der Selbstorganisation des Projekts bilden damit den Rahmen, in dem sich jede Organisation eines doppelt freien Software-Projekts abspielen muss. Wie dieser Rahmen in der Praxis gefüllt wird, ist nicht festgelegt. In sehr vielen Projekten gibt es jedoch das Maintainer-Modell.

Das Maintainer-Modell unterscheidet im Wesentlichen zwei Rollen, den oder die MaintainerIn und andere TeilnehmerInnen. Die Aufgaben der MaintainerIn bestehen im Wesentlichen darin, das Projekt generell auf Kurs zu halten. Die MaintainerIn entscheidet verbindlich über die grundsätzliche Richtung, in die die Software des Projekts weiterentwickelt werden soll, kümmert sich um die Einhaltung projektinterner Standards und darum, dass das Projekt sich bei Bedarf überhaupt weiterentwickelt. Nicht selten regelt die MaintainerIn auch die Außenkontakte für das Projekt. MaintainerInnen kommandieren jedoch nicht die anderen TeilnehmerInnen, vielmehr leisten diese freiwillig Beiträge zum Projekt in Form von Code, Dokumentation, Bug-Reports und vielem anderen mehr (Vgl. Raymond 2000).

Die speziellen Rahmenbedingungen doppelt freier Software führen zu Strukturen, die sich wesentlich von den bekannten Leitungsformen bei herkömmlicher Software–Entwicklung unterscheiden. Da die TeilnehmerInnen freiwillig am Projekt teilnehmen, können Entscheidungen nur getroffen werden, wenn der *Konsens* der wichtigen TeilnehmerInnen erreicht wird. Konsens meint hier nicht, dass alle zustimmen müssen (Einstimmigkeit), sondern Konsens ist vielmehr erreicht, wenn die TeilnehmerInnen einer Entscheidung nicht widersprechen müssen. Abstimmungen, wie sie in einigen Projekten vorgesehen sind, sind meist nur Mittel, um ein Stimmungsbild zu erzeugen.

Schafft es eine MaintainerIn in wichtigen Fragen nicht, einen Konsens herbeizuführen, so wird sie bald ohne TeilnehmerInnen da stehen. Gleichzeitig sind die TeilnehmerInnen darauf angewiesen, dass es Personen gibt, die die Aufgaben der MaintainerIn übernehmen und den Konsens organisieren. So ergibt sich eine gegenseitige Abhängigkeit zwischen MaintainerIn und anderen TeilnehmerInnen. Für das Produktivkraftmodell, das wir bei freier Software beobachten können, sind konsensorientierte Organisationsformen, wie z. B. das Maintainer-Modell, die Konflikte optimal ausbalancieren, unabdingbar.

7. Eine neue Qualität von Produktivkraftentwicklung

Eine wichtige Basis der Argumentation im Projekt Oekonux ist, dass es sich bei dem Phänomen freie Software um ein Beispiel für ein qualitativ neues Modell von *Produktivkraftentwicklung* handelt. Unter Produktivkraftentwicklung verstehen wir die historische Entwicklung der *Produktivkraft*, wobei Produktion in diesem Zusammenhang als das Stoffwechselverhältnis zwischen Mensch und (äußerer) Natur betrachtet wird. Produktivkraft fasst das Verhältnis zwischen Menschen, Produktionsmitteln und der Natur zusammen (Marx 1974).

Wir können drei Dimensionen von Produktivkraft unterscheiden. Die Dimension des *Inhalts* beschreibt, was der Inhalt menschlicher Tätigkeit ist – also die Art der Produkte, der Bezug zur Natur und die verwendeten Produktionsmittel. Im hier betrachteten Fall also freie Software und die für ihre Herstellung verwendeten Produk-

tionsmittel. Die Dimension der *Form* beschreibt die Art und Weise der Organisation des Produktionsprozesses – ob also z. B. Arbeitsteilung eingesetzt wird. Bei freier Software gehört die Selbstorganisation und das Maintainer-Modell in diesen Bereich. Die Dimension der *Produktivität* beschreibt die produzierte Gütermenge pro Zeiteinheit.

Wenn sich diese Dimensionen der Produktivkraft verändern, sprechen wir von einem qualitativen Schritt in der Produktivkraftentwicklung. Bei freier Software sehen wir eine Veränderung vor allem beim Inhalt und der Form der Produktivkraft. Im Folgenden werden einige Aspekte freier Software beschrieben, die wir für Hinweise auf diese neue Qualität halten.

Software insgesamt ist eine Produktgruppe, die erst seit einigen Jahrzehnten existiert. Ihre Nutzung setzt das Vorhandensein von Computern, mithin also hochmoderner Geräte voraus. Software, damit auch freie Software, ist also ein hochmodernes Produkt, das auf einem früheren Stand von Produktivkraftentwicklung gar keinen Sinn gemacht hätte. Freie Software befindet sich als Produkt an der *Spitze der allgemeinen Produktivkraftentwicklung*.

Nach wie vor unterliegen die Paradigmen, unter denen Software hergestellt wird, einem schnellen Wandel: Strukturierte Programmierung, Objektorientierung, Wasserfallmodell und agile Methoden – um nur ein paar zu nennen – haben sich innerhalb weniger Jahre abgelöst. Wir erleben ein Entwicklungstempo an den Wurzeln einer Technologie, das in anderen Ingenieursdisziplinen längst Vergangenheit ist. Mit einigem Recht kann freie Software als Produktionsweise ebenfalls als neues Paradigma bezeichnet werden. Einer der fundamentalen Unterschiede wird schon in der berühmt gewordenen Tanenbaum-/Torvalds-Debatte aus der Frühzeit der Linux-Entwicklung deutlich. Der Informatik-Professor Tanenbaum bezeichnete darin das von Torvalds für Linux avisierte Entwicklungsmodell als nicht praktikabel, da es nicht möglich sei, tausend Primadonnen zu kontrollieren. Die lakonische Antwort von Torvalds, die das Maintainer-Modell im Wesentlichen vorwegnimmt, bestand darin, dass es nicht seine Absicht sei, zu kontrollieren.[6] Diese Aufgabe von Kontrolle bezieht sich dabei sowohl auf die „Primadonnen" als auch auf den Code selbst. Freie Software gehört hinsichtlich seiner Produktionsweise zu einem der *innovativsten Ansätze*.

Freie Software wird nicht nur auf Computern benutzt und über das Internet verteilt, sondern auch mit Hilfe von Computern und Internet entwickelt. Computer allgemein und speziell ihre Anwendung in Form des Internets sind die zentralen Produktionsmittel für die Entwicklung freier Software. Diese Produktionsmittel gehören ebenfalls zu den *am weitesten entwickelten Produktionsmitteln*, die die Menschheit bisher hervorgebracht hat.

Im Gegensatz zu Produktionsmitteln vorangegangener Produktivkraftepochen sind Computer und das Internet auf Grund ihrer Universalität nicht auf Produktion digitaler Güter festgelegt, sondern können auch zum Spielen, zum Musikmachen, zum Diskutieren etc. eingesetzt werden. Die Produktionsmittel freier Software sind zunehmend *Teil der allgemeinen Infrastruktur* der sich am Horizont abzeichnenden Informationsgesellschaft.

6 Siehe hierfür http://www.educ.umu.se/~bjorn/mhonarc-files/obsolete/msg00089.html oder Torvalds (2001).

Diese allgemeine Infrastruktur ist heutzutage so preiswert und gleichzeitig allgemein nützlich geworden, dass sie in den hochindustrialisierten Regionen bereits beinahe überall verfügbar ist. Die Produktionsmittel, auf denen freie Software beruht, befinden sich also in *breiter privater Verfügung*. Auch dies ist ein Aspekt, der für Produktionsmittel vorangegangener Produktivkraftepochen nicht gilt.

Die Teilnahme an freien Software-Projekten ist nicht an Staaten oder Kulturkreise gebunden. Ganz selbstverständlich finden sich alle Interessierten an einem Projekt via Internet und kooperieren, um ein Produkt zu erstellen, das ihnen entspricht. Entwicklung freier Software ist *transnational*. Sie bezieht sich hinsichtlich ihrer Lizenzen auf die nationalstaatlichen Rechtssysteme der früheren Produktivkraftepoche und definiert somit einen eigenen Raum jenseits der Nationalstaaten.

Bemerkenswert ist auch, dass das gesamte Phänomen freier Software aus der Zivilgesellschaft kommt. Weder staatliche Agenturen noch Firmen haben freie Software hervorgebracht. Erst in neuerer Zeit, nachdem freie Software bereits erhebliche Erfolge erzielt hat, beginnen staatliche Einrichtungen und die Wirtschaft, auf den fahrenden Zug aufzuspringen. Freie Software würde sich auch unabhängig von diesen Einflüssen weiter entwickeln. Anstatt, dass Staat oder Wirtschaft die Kontrolle übernehmen, gibt es viele Beispiele dafür, dass sie sich den Gegebenheiten freier Software anpassen. So hat beispielsweise IBM zu Beginn seines Engagements im Bereich freier Software explizit darauf hingewiesen, dass man als großer Player behutsam mit der Community umgehen müsse – was allem Anschein nach bis heute erfolgreich durchgehalten wird und zu einer gewissen Reputation in der Community geführt hat. IBM hat verstanden, dass der Versuch eine Einschränkung der Freiheit die Kuh schlachten würde, die sie gerne melken möchte. So haben Firmen wie IBM ein großes Interesse daran, freie Software-Projekte zu unterstützen, da sie von den Leistungen der Community in höherem Maße profitieren, als sie es durch eine Kontrolle des Code erreichen würden.

8. Digitale Kopie als technologische Grundlage

Wir haben erläutert, dass bei der Entwicklung freier Software die Selbstentfaltung den individuell-sozialen Aspekt der Produktivkraftentwicklung bildet. Als technologische Seite dieser Entwicklung tritt die *digitale Kopie* hinzu. Während der Aspekt der Selbstentfaltung als gesellschaftliche Potenz schon immer da gewesen ist, handelt es sich bei der digitalen Kopie um eine historisch neue Potenz. Erst durch diesen technologischen Fortschritt ist die Ausdehnung der Selbstentfaltung als Grundlage eines Produktivkraftmodells möglich geworden.

Die digitale Kopie, die Möglichkeit also, digitale Informationen zu reproduzieren, hat im technologischen Sinne einige Eigenschaften, die sie älteren Technologien gegenüber voraus hat.

Während analoge Reproduktionen von Information immer mit Verfälschungen zu kämpfen haben, liefert die digitale Kopie eine *exakte Reproduktion* des Originals: Original und Kopie sind nicht zu unterscheiden. Mit diesem technologischen Fortschritt werden Begrenzungen der Verfügbarkeit digitaler Informationen nachhaltig beseitigt.

Zwei weitere Tatsachen moderner Technologieentwicklung geben der digitalen Kopie aber erst richtig Sprengkraft. Einerseits ist diese Reproduktionstechnik nämlich mittels Computern für sehr viele Menschen täglich und selbstverständlich verfügbar. Diese *breite Verfügbarkeit* führt dazu, dass die digitale Kopie kaum noch Einschränkungen unterliegt. Digital-Rights-Management-Technologien sind so gesehen nichts anderes als der (krampfhafte) Versuch, diese basale Eigenschaft moderner Technologie wieder zurückzunehmen, denn tatsächlich sind die beiden fundamentalen Operationen von Computern die Manipulation und die Kopie von digitalen Daten.

Andererseits steht mit dem Internet, das nichts anderes als eine planetenumspannende Fernkopiereinrichtung ist, eine Einrichtung zur Verfügung, die es ermöglicht, dass Informationsgüter auf einfachste Weise *global* verfügbar gemacht werden können. Das Internet verbindet die individuelle Verfügung über Informationsgüter mit dem allgemeinen Zugang zu ihnen.

Ein weiterer, eher subtiler Aspekt digitaler Kopie ist ihre *Universalität*: Der Inhalt, die Bedeutung des zu kopierenden Informationsguts, ist für den Vorgang der digitalen Kopie völlig unerheblich. Texte, Bilder, Musik, Programme können mit der gleichen Technologie reproduziert werden, sobald sie als Bytestrom vorliegen. So, wie die Kraftmaschinen der industriellen Ära (Dampfmaschine, vor allem aber Elektromotor) eine Basistechnologie für beliebige Anwendungen mechanischer Kraft und damit für die Industriegesellschaft bilden, so bildet die digitale Kopie eine Basistechnologie für die Informationsgesellschaft.

Auf dieser Grundlage ist das Internet von Beginn an auch als Kommunikationsmittel genutzt worden. Wie keine Kommunikationseinrichtung zuvor ermöglicht das Internet *globale Kommunikation* in Echtzeit. Es ist jetzt möglich, dass Menschen mit gleichen Bedürfnissen unabhängig von ihrem Standort in dem Tempo kommunizieren, das ihnen und ihrer Tätigkeit angemessen ist. Diese Kooperationsmöglichkeit ist wie die digitale Kopie selbst eine unabdingbare Voraussetzung für die Entfaltung freier Software.

9. Freie Software als Keimform

Eine der zentralen und nicht unumstrittenen Thesen im Projekt Oekonux ist die These von der freien Software als *Keimform einer neuen Gesellschaft* (Vgl. Kurz 1997). Unter einer Keimform verstehen wir ein Phänomen, das in den Rahmen eines bestehenden Gesamtsystems eingebettet ist, gleichzeitig aber Eigenschaften hat, die über die Logik des umgebenden Gesamtsystems hinausgehen und eine mögliche, neue Entwicklungsrichtung des Gesamtsystems darstellen können. Eine Keimform ist dabei noch keine vollständig entfaltete Form einer Gesellschaft, sondern zeigt nur einige identifizierbare Merkmale.

Eine neue Form entfaltet sich dagegen in mehreren Schritten. Das so genannte *Fünfschritt-Modell*, aus dem der Begriff Keimform stammt, erfasst in allgemeiner Weise, wie es innerhalb von Entwicklungsprozessen zu qualitativen Übergängen kommt. Es erklärt, wie Neues entsteht und sich schließlich durchsetzen kann. Das Fünfschritt–

Modell kommt ursprünglich aus der Kritischen Psychologie (Holzkamp 1983), aus der Analyse qualitativer Entwicklungsschritte in der Evolution. Die fünf Schritte sind:

1. Entstehung der Keimform
   Alles, was es selbstverständlich und allgegenwärtig gibt, ist irgendwann einmal etwas Neues, ganz und gar nicht Selbstverständliches gewesen. Über mehrere Schritte hat sich das Neue schließlich durchgesetzt. Dieses Neue, das später einmal Altes sein wird, nennt man Keimform. Keimformen können in Nischen und Sonderbereichen entstehen. Sie leben vom und im Alten, besitzen aber schon Formen des Neuen.

2. Krise der alten Form
   Keimformen erlangen nur Bedeutung, wenn das Alte in die Krise gerät. Das Alte kann im Wesentlichen aus zwei Gründen in die Krise geraten. Zum einen können sich äußere Bedingungen so dramatisch oder so schnell verändern, dass das alte Prinzip darauf nicht mehr angemessen reagieren kann. Zum anderen kann sich das Alte selbst erschöpft haben, wenn alle Entwicklungspotenzen ausgereizt sind. Stagnation wäre eine Reaktionsform, Zerfall eine andere.

3. Keimform wird zur wichtigen Entwicklungsdimension
   Unter den Bedingungen der Krise des Alten kann die Keimform die Nischen verlassen und sich quantitativ ausbreiten. Sie wird zu einer wichtigen Entwicklungsdimension innerhalb der noch dominanten alten Form. Diese Etablierung der Keimform kann zwei Richtungen einschlagen: Sie führt zur Integration in das Alte und zur Übernahme der alten Prinzipien oder die Keimform behauptet sich auf Grund der neuen Prinzipien immer besser im und neben dem Alten. Im ersten Fall geht der Keimformcharakter verloren, im zweiten Fall wird das Neue gestärkt. Das Alte kann in beiden Fällen von einer integrierten oder gestärkten Keimform profitieren und Krisenerscheinungen abmildern.

4. Keimform wird zur dominanten Größe
   Die frühere minoritäre Keimform wird zur dominanten Form der Entwicklung. Das Neue setzt sich durch, weil es hinsichtlich einer wichtigen Dimension des Gesamtprozesses besser ist. Damit endet der Keimformcharakter des Neuen. Nun sind seine Prinzipien bestimmend und verdrängen nach und nach oder auch schlagartig die überkommenen, nicht mehr funktionalen Prinzipien des Alten. Das Neue wird das selbstverständliche Allgegenwärtige.

5. Umstrukturierung des Gesamtprozesses
   Schließlich strukturieren sich alle Aspekte des Gesamtprozesses in Bezug auf das bestimmende, jetzt selbstverständliche Neue hin um. Das betrifft vor allem auch solche Prozesse, die im Gesamtprozess nicht bestimmend, sondern nur abgeleitet sind. Mit diesem Schritt ist nun potenziell wieder der erste Schritt eines neuen Fünfschrittes erreicht: Keimformen können auftreten, das dann alte Neue gerät in die Krise usw.

Alle Phasen können über kürzere oder längere Zeiträume andauern und es kann jederzeit Rückschritte geben. Nichts ist vorgegeben oder determiniert. Vollständig begriffen kann ein Fünfschritt der Entwicklung erst werden, wenn er vollzogen wurde und erst im Nachhinein kann man die frühere Keimform sicher identifizieren. Mitten im Entwicklungsprozess begriffen kann das Fünfschrittmodell helfen, die Sinne zu schärfen, um handlungsfähiger zu werden. Die umstrittene These im Projekt Oekonux lautet nun: „Bei freier Software haben wir es mit einer Keimform einer neuen Gesellschaft zu tun".

Wie beschrieben, zeichnet sich freie Software durch Wertfreiheit, Selbstentfaltung, Selbstorganisation und Globalität aus. Das alte Prinzip der Warengesellschaft basiert demgegenüber auf dem Wertgesetz, der Selbstverwertung, der Entfremdung und den Nationalstaaten. Die alte Form, die Warengesellschaft, ist erkennbar in der Krise. So versprach noch in den siebziger Jahren des vergangenen Jahrhunderts eine weitere Entfaltung und Vertiefung der Warengesellschaft nicht nur den Menschen der hoch industrialisierten Staaten individuelle Wohlstandsmehrung, sondern dieses Versprechen konnte auch erkennbar eingehalten werden. Von einem Fortschritt in dieser Hinsicht spricht heute niemand mehr. Vielmehr wird im Rahmen von Arbeitslosigkeit und Sozialabbau ganz offen und in steigendem Tempo nur noch darüber diskutiert, wie die Senkung des Lebensstandards breiter Bevölkerungskreise auch der hoch industrialisierten Staaten weiter vorangetrieben werden soll. Demgegenüber hat sich freie Software als neue Form von Reichtum etabliert, das jenseits der Formen der Verwertung existiert. Umstritten ist, ob freie Software bereits eine wichtige Entwicklungsdimension innerhalb der alten Form (dritter Schritt) geworden ist oder sich noch in einer der früheren Phasen befindet.

10. GPL-Gesellschaft

In der Diskussion um die gesellschaftlichen Potenzen des Entwicklungsmodells, das sich nach der These in freier Software keimförmig zeigt, kam der Begriff der GPL–Gesellschaft auf (Merten 2000). Er bezeichnet eine mögliche zukünftige Gesellschaftsform, die auf den Prinzipien der Entwicklung freier Software beruht. Es ist aus verschiedenen Gründen nicht seriös möglich, ein detailliertes Bild einer solchen Vorstellung zu entwerfen. Entlang der Prinzipien der Entwicklung freier Software lassen sich aber einige Rahmenelemente feststellen.

Ein wichtiges Element der Entwicklung freier Software ist die Tatsache, dass die verwendeten Produktionsmittel vergleichsweise vielen Menschen Selbstentfaltung ermöglichen. Der innere Grund dafür ist, dass Computer als universelle, Information verarbeitende und programmierbare Maschinen *unendlich viele Freiheitsgrade* haben. Diese Freiheitsgrade können von Menschen zur Entfaltung ihrer individuellen Kreativität genutzt werden.

In einer GPL-Gesellschaft wäre diese Eigenschaft tendenziell auf alle Produktionsmittel übertragbar. Dies bedeutet, dass der Maschinenpark, den die Industriegesellschaft für die Nutzung unter den entfremdeten Bedingungen der Lohnarbeit hervorgebracht hat, umgearbeitet oder neu entworfen werden muss: Die Arbeit an

einem Fließband dürfte beispielsweise nur für die allerwenigsten Menschen zur Selbstentfaltung führen, weswegen sie endgültig verschwinden müsste.[7]

Auch die weitere und noch beschleunigte *Automatisierung von Arbeitsprozessen* ist ein Mittel, um maximale Selbstentfaltung zu gewährleisten. Arbeitsprozesse, die im Kern von Maschinen übernommen werden, müssen nicht mehr von Menschen erledigt werden. Sie können sich den Aufgaben zuwenden, die genuin menschliche Fähigkeiten erfordern und nicht von Maschinen übernommen werden können.

Wie wenige andere Beispiele macht freie Software sichtbar, dass Selbstentfaltung nicht sinn- und zweckfreies Tun sein muss, wie es uns die Freizeitindustrie weismachen will. Vielmehr ist das Ergebnis der Entwicklung freier Software ein Produkt, das für viele Menschen nützlich ist. Selbstentfaltung, wie sie in freier Software praktiziert wird, hat also nicht nur für das Individuum eine positive Funktion, sondern nutzt der gesamten Gesellschaft. In einer GPL-Gesellschaft hätten noch sehr viel mehr Tätigkeiten diesen Charakter der *unmittelbaren Verknüpfung von individuellem und gesellschaftlichem Nutzen*.

Nicht zu vergessen ist, dass existierende freie Software frei verfügbar ist. Diese Eigenschaft ist einerseits eine Folge des offenen Entwicklungsprinzips, das die maximale Inklusion aller Interessierten zum Ziel hat. Unter den Bedingungen der universellen digitalen Kopierbarkeit führt dies zusammen mit dem *Copyleft* tendenziell zu allgemeiner freier Verfügbarkeit der Produkte. Andererseits ist diese freie Verfügbarkeit auch Voraussetzung für die blühende freie Software-Landschaft, denn auch die EntwicklerInnen von freier Software setzen auf von anderen entwickelter freier Software auf.

Die freie Verfügbarkeit ist also sowohl Folge als auch Voraussetzung des gesamten Entwicklungsmodells. Diese enge Verschränkung wäre in einer GPL-Gesellschaft ausgedehnt auf alle Informationsgüter sowie auf materielle Güter.

In einer GPL-Gesellschaft wären folglich viele Einrichtungen der Arbeitsgesellschaft überflüssig. Wo Güter frei verfügbar sind, ist die Form der Ware nicht mehr zu halten, die davon lebt, dass Güter künstlich verknappt werden. Werden keine Waren mehr – wohl aber Güter – produziert, so ist auch kein Geld mehr notwendig, das die Vergleichbarkeit von Waren vermittelt: Wo Güter frei zur Verfügung stehen, ist der Tausch eines Guts gegen ein anderes zur überflüssigen Handlung geworden. Nicht zuletzt würde unter den Bedingungen der GPL-Gesellschaft Entfremdungspotential an vielen Stellen tendenziell abgeschafft. Die wichtigste Produktivkraft einer GPL-Gesellschaft wäre die menschliche Selbstentfaltung.

In einer GPL-Gesellschaft würde die Selbstentfaltung der Individuen zur unmittelbaren Voraussetzung für die Selbstentfaltung aller: Nur wenn sich die Individuen entfalten können, entstehen Produkte, die für alle nützlich sind. Gleichzeitig sind diese nützlichen Produkte und deren freie Verfügbarkeit die Grundlage für die individuelle Tätigkeit: Die Selbstentfaltung aller ist also auch die Voraussetzung für die Selbst-

7 Bereits in der Industriegesellschaft gibt es verschiedene Versuche, die Selbstentfaltung in die Produktion zu integrieren und so die Kreativitätsreserven der Lohnabhängigen zu mobilisieren. Allerdings können diese Versuche nicht die strukturelle Schranke der Entfremdung des Arbeitsprozesses überwinden (Vgl. auch Gorz 2004).

entfaltung der Individuen. Wir haben es mit einem sich selbst verstärkenden Prozess zu tun, der die langfristige Tragfähigkeit einer GPL-Gesellschaft in einem günstigen Licht erscheinen lässt.

11. Historische Schritte im Vergleich

Wenn wir davon ausgehen, dass freie Software ein Hinweis auf einen fundamentalen Schritt in der Produktivkraftentwicklung ist, so kann ein Vergleich mit dem letzten fundamentalen Schritt in der Produktivkraftentwicklung interessante Einsichten geben. Die letzte fundamentale Änderung war der Umbruch von den feudal geprägten Gesellschaften des Spätmittelalters zu den industriell geprägten Gesellschaften der Neuzeit mit Beginn der Aufklärung. Die gerade erfundenen Industriemaschinen erforderten für ihren Betrieb hinsichtlich Technik und Organisation von Menschen eine völlig neue Produktionsweise. Dies verwob sich mit sozialen und ideologischen Entwicklungen, wie beispielsweise der Idee der Nationalstaaten, sodass innerhalb eines historisch relativ kurzen Zeitraums diese Entwicklung der Produktivkräfte zu einer tiefgreifenden Umstrukturierung der Gesellschaft führte.

Einerseits wurden damals menschliche, tierische und einige wenige natürliche Kraftquellen (Wasser, Wind) durch moderne Kraftquellen (Dampf, Elektrizität) abgelöst. Andererseits wanderte das Know-how über die Arbeitsprozesse von den sie ausführenden Menschen in die mechanische Konstruktion der Maschinen, so dass menschliche Arbeitskraft nur noch als Ergänzung zu den Maschinen benötigt wurde. Das Ergebnis dieser Umstellung waren eine Fülle neuer, nützlicher Produkte sowie Großprojekte, die ohne Einsatz industrieller Methoden nicht denkbar gewesen wären. Eine unrühmliche Antriebsfeder war die industrielle Entwicklung von Militärtechnik.

Gleichzeitig strukturierte sich die Lebensweise der Menschen tiefgreifend um. Die Art und Weise, wie Menschen in der feudalen, subsistenzorientierten Produktionsweise ihr Leben organisiert haben, war für die industrielle Produktionsweise in vielerlei Hinsicht ungeeignet. Um nur ein Beispiel herauszugreifen: Der vorindustrielle Umgang mit Zeit war weitgehend an den Hell-/Dunkelphasen und den konkreten, unmittelbaren eigenen Notwendigkeiten orientiert. Dies ging so weit, dass in manchen Klöstern die Länge einer Stunde über den Jahreslauf variierte. Für die industrielle Produktion, bei der der Zeittakt durch die Maschinen vorgegeben wird, waren das völlig unbrauchbare Verhältnisse. Es soll nicht verschwiegen werden, dass diese Umstrukturierung der Lebensweise durchaus nicht immer freiwillig geschah, sondern in erheblichem Ausmaß auch mit dem Einsatz von Gewalt einherging.

Auch beim Übergang von der feudalen zur bürgerlichen Gesellschaftsform lassen sich Keimformen ausmachen. So kann beispielsweise das Handelskapital, das sich bereits im Frühmittelalter auszubilden begann, als frühe Keimform betrachtet werden, bei der sich der kapitalistische Umgang mit Geld entwickelte. Die frühindustrielle Textilindustrie war für diesen Schritt in der Produktivkraftentwicklung die Keimform, in der schon viele Formen der späteren Industriearbeit auftauchten. Insbesondere traten hier sowohl die standardisierte Massenproduktion als auch der massenhafte Einsatz von bezahlten Arbeitskräften auf.

Es lässt sich an diesem Beispiel auch sehr schön betrachten, wie die feudalen Strukturen, die alten dominanten Strukturen also, sich an diese Entwicklung anpassten und sie auch nutzten. Die gesamte Kriegsproduktion, die die Fürsten für die Führung ihrer Kriege brauchten, hatte sich von der subsistenzorientierten Produktion über die Jahrhunderte hin vollständig entbettet. Die Kriegsproduktion der späten Feudalherren, in der auch industrielle Formen relativ früh auftauchten, war nur mit Geld überhaupt zu bewerkstelligen. Das dafür notwendige Steuersystem trieb das Geldwesen weiter an, das sich später als eine entscheidende Grundlage nach-feudaler Gesellschaftsformen herausstellen sollte. Söldnerwesen sowie stehende Heere können als frühe Form bezahlter „Arbeitskraft" betrachtet werden.

Heute wissen wir, dass diese Entwicklung, die in frühen Keimformen bereits erkennbar war, zu einer grundlegenden Umwandlung der Gesellschaftsform geführt hat. Auch wenn die Fürsten die Aufklärung und die mit ihr verbundende, industrielle Produktionsweise teilweise begrüßten oder sogar vorantrieben, so stellte sich doch heraus, dass die feudale Produktionsweise mit ihrem Privilegiensystem und der Leibeigenschaft industrieller Produktion unangemessen war. Die industrielle Produktionsweise musste sich also neue Grundlagen schaffen, die durch Geldwirtschaft und freie Lohnarbeit gekennzeichnet waren. Auch die gesamte gesellschaftliche Organisation folgte nach und nach dieser Entwicklung der Produktivkraft – am sichtbarsten in der Gründung bürgerlicher Nationalstaaten.

Heute sind wir an einem Punkt angekommen, wo sich die mechanische Konstruktion von Maschinen immer weiter von einem konkreten Arbeitszweck ablöst. Industrieroboter, Fabber[8] etc. sind nicht durch ihre Konstruktion auf einen bestimmten Arbeitsprozess festgelegt, sondern ihre mechanische Konstruktion steckt nur noch den Rahmen ihrer Möglichkeiten ab. Die Produktion konkreter Gegenstände ist bei diesen Maschinen bereits Gegenstand von Software, womit das Know-how über die Arbeitsprozesse zum Informationsgut wird. Computer verwenden diese Informationsgüter als Programme in automatisierten Prozessen, sodass menschliche Energie und Kreativität nur noch dafür benötigt wird, die Informationsgüter selbst zu erstellen.

Die Nutzung von Industriemaschinen erforderte auf Grund ihrer Beschränkungen eine Anpassung der Menschen an die Notwendigkeiten der Maschinerie, was in vielerlei Hinsicht letztlich eine Unterwerfung bedeutete. Die Produktion von Informationsgütern ist hingegen ein kreativer Prozess, bei dem gerade die schöpferischen Qualitäten des Menschen gefragt sind, die durch Unterwerfung vernichtet werden. Zog die beginnende Industriegesellschaft eine Unterwerfung der Menschen nach sich, so erfordert die beginnende Informationsgesellschaft eine Freisetzung der unbeschränkten Selbstentfaltung von Menschen.

Während sich beim Übergang von den feudalen zu den bürgerlichen Gesellschaften der Schwerpunkt der Produktion von der Nutzung des Bodens zur industriellen Produktion materieller Güter verlagerte, so verschiebt sich der Schwerpunkt der

8 Ein Fabber (*digital fabricator*) wird auch als „Fabrik in einer Kiste" bezeichnet. Fabber erzeugen dreidimensionale Werkstücke direkt aus digitalen Daten, z. B. aus Kunststoff oder Metall. Fabber werden derzeit hauptsächlich im „Rapid Prototyping" zur Fertigung von Prototypen und Modellen verwendet.

Produktion beim Übergang in die *heraufziehende Informationsgesellschaft* auf die Produktion neuer Informationsgüter. Der Wechsel zur Industrieproduktion erforderte einen fundamentalen Wechsel in der Gesellschaftsform. Informationsgüter, die in fast allen Aspekten anderen Bedingungen unterliegen als materielle Güter, erfordern eine ebensolche Umstrukturierung. Eine grundlegende Änderung der Gesellschaftsform erscheint unabdingbar.

Literaturverzeichnis

Gorz, A. (2004), *Wissen, Wert und Kapital. Zur Kritik der Wissensökonomie*, Rotpunktverlag, Zürich.

Himanen, P. (2001), *The Hacker Ethic and the Spirit of the Information Age*, 1. Aufl., Random House, Vintage, UK.

Holzkamp, K. (1983), *Grundlegung der Psychologie*, Campus Verlag, Frankfurt am Main.

Kurz, R. (1997), 'Antiökonomie und Antipolitik. Zur Reformulierung der sozialen Emanzipation nach dem Ende des „Marxismus"', *Krisis* **19**. http://www.giga.or.at/others/krisis/r-kurz_antioekonomie-und-antipolitik_krisis19_1997.html.

Marx, K. (1974), *Grundrisse der Kritik der politischen Ökonomie*, Dietz-Verlag, Berlin/DDR. Rohentwurf 1857–1858.

Meretz, S. (2000*a*), 'GNU/Linux ist nichts wert – und das ist gut so!', http://www.kritische-informatik.de/?lxwertl.htm.

Meretz, S. (2000*b*), *Linux Co. Freie Software – Ideen für eine andere Gesellschaft*, AG SPAK Verlag, Neu-Ulm.

Meretz, S. (2004), 'Freie Software. Über die Potenziale einer neuen Produktionsweise', *Widerspruch* **45**.

Merten, S. (2000), GNU/Linux – Meilenstein auf dem Weg in die GPL-Gesellschaft, *in* 'Dokumentation LinuxTag 2000'. http://www.oekonux.de/texte/meilenstein/.

Raymond, E. S. (2000), 'The Cathedral and the Bazaar', http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/.

Torvalds, L. (2001), *Just for Fun – Wie ein Freak die Computerwelt revolutionierte*, Carl Hanser Verlag.