

# "Virtuelles" Eigentum?

Referat auf der Konferenz "RechtsLinks"  
der Vereinigung Demokratischer Juristinnen und Juristen (VDJ)  
Berlin, 10. November 2001

Dipl.-Inform. Robert A. Gehring<sup>1</sup>

## 0. Einleitung

Wie nähert man sich am besten der Frage nach der Bedeutung von Eigentum im Internet - "virtuellem" Eigentum also -, wenn einem dafür nur zwanzig Minuten Zeit zur Verfügung stehen? Vielleicht sollte man auf Details verzichten? Vielleicht sollte man einen philosophischen Zugang zur Materie wählen? Vielleicht *so* beginnen, *mit Hegel*?

*«Das Eigentümliche an der geistigen Produktion kann durch die Art und Weise der Äußerung unmittelbar in solche Äußerlichkeit einer Sache umschlagen, die nun ebenso von anderen produziert werden kann; so daß mit deren Erwerb der nunmehrige Eigentümer, außerdem daß er damit die mitgeteilten Gedanken oder die technische Erfindung zu eigen machen kann, welche Möglichkeit zum Teil (bei schriftstellerischen Werken) die einzige Bestimmung und den Wert des Erwerbs ausmacht, zugleich in den Besitz der allgemeinen Art und Weise, sich so zu äußern und solche Sachen vielfältig hervorzubringen, kommt.»*

Das Zitat stammt aus den **Grundlinien der Philosophie des Rechts**, §68 (Hegel 1821: 74), worin sich Hegel u.a. mit Fragen des geistigen Eigentums auseinandersetzt.

Nun kann die Lektüre von Hegel ohne Zweifel sehr erhellend sein. Ob man mit ihrer Hilfe den Vorgängen im Internet auf die Spur kommen kann, wage ich jedoch zu bezweifeln. Ich nehme Abstand

---

<sup>1</sup> E-Mail: <mailto:rag@cs.tu-berlin.de>

Der vorliegende Text ist im Internet über <http://ig.cs.tu-berlin.de/ap/rg/index.html> verfügbar.

von Hegel und verzichte an dieser Stelle auch auf andere eher metaphysische Ansätze, obwohl geeignete Kandidaten in Fülle zur Verfügung stünden.

Stattdessen besinne ich mich darauf, daß mein professionelles Arbeitsfeld die angewandte Informatik ist. Das Internet -als *das* Beispiel *par excellence* für die Anwendung von Informatik- könnte, aus solcher Perspektive betrachtet, verständlicher werden. Und so habe ich mich für einen empirischen Zugang zur Problematik entschieden.

Bevor die Stärken und Schwächen einer Institution wie der des Eigentums unter den Bedingungen des Internets herausgearbeitet werden können, stelle ich die Frage, was denn diese *Bedingungen des Internets* sein können. Was ist das Besondere am Internet, das uns drängt, die Frage nach dem Eigentum zu stellen?

Diese Suche nach einer Antwort auf *diese Frage* kann unseren Blick auf das Phänomen der "freien Software" lenken, entzündet sich doch daran regelmäßig die Debatte.

Im folgenden möchte ich mich -in der gebotenen Kürze- drei Kernfragen widmen:

1. **Was** ist freie Software?

Ich gehe davon aus, daß noch nicht jeder der Anwesenden eine detaillierte Vorstellung davon hat. Es könnte deshalb hilfreich sein, ganz konkret zu werden.

2. **Wer** entwickelt eigentlich freie Software?

Stimmen die gängigen Mythen, die uns das Bild von langhaarigen, leicht übergewichtigen und schwer übernachtigten Teenager-Hackern vermitteln wollen?

3. Die Frage nach dem **Warum** soll gestellt werden.

Was veranlaßt die Entwickler, die Ergebnisse ihrer Tätigkeit frei zur Verfügung zu stellen? Ist es Altruismus, wie oft kolportiert wird? Oder ist es etwa bloß die Absicht, dem Giganten Microsoft "*eins auszuwischen*"?

Ich hoffe, daß nach meiner Betrachtung dieser drei Punkte etwas klarer geworden ist, welche Bedeutung der Frage nach dem Eigentum im Internet zukommt.

Ich beginne mit dem ersten Spotlight, der Frage danach, was eigentlich freie Software ist.

## **1. Am Anfang ist der Code**

Auf dem gezeigten Bild sehen Sie ein "echtes" Stück freie Software. Es handelt sich um einen Ausschnitt aus dem freien Betriebssystem Linux, das der einen oder dem anderen unter Ihnen vielleicht bekannt ist.

```

/*
 * linux/kernel/sched.c
 *
 * Kernel scheduler and related syscalls
 *
 * Copyright (C) 1991, 1992 Linus Torvalds
 *
 * 1996-12-23 Modified by Dave Grothe to fix bugs in semaphores and
 *           make semaphores SMP safe
 * 1998-11-19 Implemented schedule_timeout() and related stuff
 *           by Andrea Arcangeli
 * 1998-12-28 Implemented better SMP scheduling by Ingo Molnar
 */

/*
 * 'sched.c' is the main kernel file. It contains scheduling primitives
 * (sleep_on, wakeup, schedule etc) as well as a number of simple system
 * call functions (type getpid()), which just extract a field from
 * current-task
 */

#include <linux/config.h>
#include <linux/mm.h>
#include <linux/init.h>
#include <linux/smp_lock.h>
#include <linux/interrupt.h>
#include <linux/kernel_stat.h>

#include <asm/uaccess.h>
#include <asm/mmu_context.h>

extern void timer_bh(void);
extern void tqueue_bh(void);
extern void immediate_bh(void);

/*
 * scheduler variables
 */

unsigned securebits = SECUREBITS_DEFAULT; /* systemwide security settings */

extern void mem_use(void);

/*
 * Scheduling quanta.
 *
 * NOTE! The unix "nice" value influences how long a process
 * gets. The nice value ranges from -20 to +19, where a -20
 * is a "high-priority" task, and a "+10" is a low-priority
 * task.
 *
 * We want the time-slice to be around 50ms or so, so this
 * calculation depends on the value of HZ.
 */
#if HZ < 200
#define TICK_SCALE(x) ((x) >> 2)
#elif HZ < 400
#define TICK_SCALE(x) ((x) >> 1)
#elif HZ < 800
#define TICK_SCALE(x) (x)
#elif HZ < 1600
#define TICK_SCALE(x) ((x) << 1)
#else
#define TICK_SCALE(x) ((x) << 2)
#endif

```

Was man auf dem Bild sieht, ist ein Teil des Linux-Betriebssystemkerns, der von verschiedenen Programmierern in Kooperation entwickelt wurde. Das Programmmodul wurde in der Programmiersprache C geschrieben, einer Art *lingua franca* der Informatik. Die Darstellungsweise ist die des sogenannten *Quelltextes* (oder *Quellcodes*). Das Linux-Betriebssystem besteht aus einer Vielzahl solcher Module.<sup>2</sup>

Sie können eine Reihe von Kommentaren sehen, im Bild grau hinterlegt, die neben einer kurzen Funktionsbeschreibung Namen und Hinweise auf die Beiträge der einzelnen Entwickler enthalten. Sie sehen, daß eine ganze Reihe von Programmierern am Werke waren: *Linus Torvalds*, *Dave Grothe*, *Andrea Arcangeli* und *Ingo Molnar*.<sup>3</sup>

Angenommen, Sie als Anwender stellen eines Tages fest, daß in diesem Programmteil ein Fehler steckt. Dann können Sie über eine Suchmaschine -z.B. Google<sup>4</sup>- die e-Mail-Adressen ermitteln und sich direkt an einen der Entwickler wenden. Kein Callcenter-Mitarbeiter wird dafür bezahlt, Sie abzuwimmeln. Eine kurzer Test (mit Google) zeigt beispielsweise, daß etwa *Andrea Arcangeli* derzeit vermutlich unter *andrea@suse.de* erreichbar ist.

Anhand der Datumsangaben können Sie ersehen, wie so ein Stück Software im Laufe der Jahre "reift". Immer, wenn es nötig ist, wird es überarbeitet, angepaßt und um Fehler bereinigt. Unseren Beispielcode hat ursprünglich *Linus Thorvalds* entwickelt. Ende 1996 hat *Dave Grothe* Fehlerbereinigungen durchgeführt und den Code an Computer mit mehr als einem Prozessor<sup>5</sup> angepaßt. *Andrea Arcangeli* hat Ende 1998 diverse Erweiterungen vorgenommen und *Ingo Molnar* hat kurz darauf Verbesserungen für den Betrieb mit mehreren Prozessoren integriert.

Was Sie erkennen können ist, daß die Software grundsätzlich *lesbar* ist. Selbst diejenigen, die nicht programmieren können, sind in der Lage, einzelne Teile des Quelltextes zu interpretieren. Dazu gehören neben den Kommentaren auch die Benennungen mancher Variablen. Der textuelle Charakter von Software, so wie sie von den Entwicklern erfahren wird, ist evident.

---

<sup>2</sup> Um daraus ein ablauffähiges Programm zu machen, muß man zuerst den Quelltext von einem *Compiler* in die Sprache des Mikroprozessors übersetzen lassen. Anschließend werden die zusammengehörigen Programmmodule durch einen *Linker* zum ausführbaren Programm zusammengefügt.

<sup>3</sup> Anhand der Namen kann man mutmaßen, daß die Entwickler aus verschiedenen Ländern stammen.

<sup>4</sup> <http://www.google.com>

<sup>5</sup> SMP steht für *Symmetrical Multi-Processing*.

Und schließlich können Sie eine für Juristen mit Sicherheit besonders interessante Information erkennen: Den *Copyright-Hinweis* von Linus Torvalds (am oberen Rand). Ganz im Gegensatz zu dem, was mancherorten behauptet wird, verzichten die Entwickler dieser Software nicht auf den gesetzlichen Urheberrechtsschutz. Sie reklamieren ihn vielmehr explizit für sich.<sup>6</sup>

Über die Lizenzbedingen erfährt man aus diesem Code-Modul jedoch nichts. Informationen darüber werden mit dem kompletten Paket verteilt, das alle zusammengehörigen Module enthält. Im Fall von Linux handelt es sich konkret um die GPL, die GNU General Public License der Free Software Foundation. Darin wird im Detail beschrieben, was Sie mit der Software machen dürfen, und was nicht.

Ohne groß auf die Einzelheiten eingehen zu wollen, möchte ich nur einige, wesentliche Bestimmungen erwähnen:<sup>7</sup>

- \* Sie dürfen die Software beliebig oft kopieren. («*You may copy and distribute verbatim copies of the Program's source code ...*»)
- \* Sie dürfen die Software verändern. («*You may modify your copy or copies of the Program or any portion of it ...*»)
- \* Sie dürfen die kopierte Software weitergeben. («*You may copy and distribute the Program ...*»)

Der Vertrieb von freier Software in der gezeigten Form des Quellcodes liefert Ihnen die faktische Basis, die in der Lizenz bestimmten Rechte auch ausüben zu können.

Um die Bedeutung dessen noch stärker hervorzuheben, möchte ich Ihnen als zweites Bild einen Ausschnitt aus einem Microsoft-Betriebssystem<sup>8</sup> zeigen.

---

<sup>6</sup> Allerdings machen sie davon in der Regel einen anderen Gebrauch als etwa proprietäre Softwarehersteller. Statt exklusive Rechte auszuüben, erklären sie den Verzicht auf den Anspruch zur exklusiven Rechtsausübung. Sie machen aus der Software ein öffentliches Gut, statt eine Ware.

<sup>7</sup> Aus der GNU GENERAL PUBLIC LICENSE, Version 2, June 1991.

<sup>8</sup> Es handelt sich also um *proprietäre* Software.

```

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...5...
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000 MZ.....
00000010: b800 0000 0000 0000 4000 0000 0000 0000 .....@.....
00000020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000030: 0000 0000 0000 0000 0000 0000 8000 0000 .....
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468 .....!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320 t be run in DOS
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000 mode....$.
00000080: 5045 0000 4c01 1000 9099 1f37 0000 0000 PE..L.....7....
00000090: 0000 0000 e000 0e01 0b01 0500 6007 0100 .....`....
000000a0: 6039 0000 0000 0000 5032 0100 0004 0000 `9.....P2.....
000000b0: 7051 0000 0000 0100 1000 0000 1000 0000 pQ.....
000000c0: 0400 0000 0000 0000 0100 0000 0000 0000 .....
000000d0: c044 0100 0004 0000 4ddf 0100 0100 0020 .D.....M.....
000000e0: 0000 1000 0010 0000 0000 1000 0010 0000 .....
000000f0: 0000 0000 1000 0000 0000 0000 0000 0000 .....
00000100: a05b 0000 3c00 0000 9032 0100 9403 0000 .[...<....2.....
00000110: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000120: 3036 0100 300e 0000 0000 0000 0000 0000 06..0.....
00000130: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000140: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000150: 0000 0000 0000 0000 b85c 0000 dc00 0000 .....\.
00000160: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000170: 0000 0000 0000 0000 5f4c 5445 5854 0000 ....._LTEXT..
00000180: 8602 0000 0004 0000 9002 0000 0004 0000 .....
00000190: 0000 0000 0000 0000 0000 0000 2000 0068 ..... .h
000001a0: 2e74 6578 7400 0000 8442 0000 9006 0000 .text....B.....
000001b0: 9042 0000 9006 0000 0000 0000 0000 0000 .B.....
000001c0: 0000 0000 2000 0068 5f50 5445 5854 0000 .... .h_PTEXT..
000001d0: 7a00 0000 2049 0000 8000 0000 2049 0000 z... I..... I..
000001e0: 0000 0000 0000 0000 0000 0000 2000 0068 ..... .h
000001f0: 5733 3200 0000 0000 0b00 0000 a049 0000 W32.....I..
00000200: 1000 0000 a049 0000 0000 0000 0000 0000 ....I.....
00000210: 0000 0000 2000 0068 504e 5000 0000 0000 .... .hPNP.....
00000220: 9e07 0000 b049 0000 a007 0000 b049 0000 ....I.....I..
00000230: 0000 0000 0000 0000 0000 0000 2000 0068 ..... .h
00000240: 5f4c 4441 5441 0000 1c00 0000 5051 0000 _LDATA.....PQ..
00000250: 2000 0000 5051 0000 0000 0000 0000 0000 ...PQ.....
00000260: 0000 0000 2000 0068 2e72 6461 7461 0000 .... .h.rdata..
00000270: 6000 0000 7051 0000 6000 0000 7051 0000 `...pQ...`...pQ..
00000280: 0000 0000 0000 0000 0000 0000 4000 0048 .....@..H
00000290: 2e64 6174 6100 0000 c409 0000 d051 0000 .data.....Q..
000002a0: d009 0000 d051 0000 0000 0000 0000 0000 .....Q.....
000002b0: 0000 0000 4000 00c8 2e69 6461 7461 0000 ....@....idata..
000002c0: 5c06 0000 a05b 0000 6006 0000 a05b 0000 \....[...`....[...
000002d0: 0000 0000 0000 0000 0000 0000 4000 00c8 .....@....
000002e0: 5f4c 4441 5441 0000 d400 0000 0062 0000 _LDATA.....b..
000002f0: e000 0000 0062 0000 0000 0000 0000 0000 ....b.....
00000300: 0000 0000 4000 00c8 5f50 4441 5441 0000 ....@..._PDATA..
00000310: 0400 0000 e062 0000 1000 0000 e062 0000 ....b.....b..

```

Sie sehen einen Ausschnitt aus dem Betriebssystem Windows 98, Second Edition, genauer gesagt aus der Datei ACPI .SYS<sup>9</sup>.

<sup>9</sup> C:\WINDOWS\SYSTEM\ACPI.SYS

In der linken Spalte sehen Sie eine Anzahl von Adressen. Die mittleren acht Spalten beinhalten den Programmcode -in hexadezimaler Darstellung<sup>10</sup>- und die rechte Spalte gibt den Code in ASCII-Darstellung wieder.

Wie Sie sich selbst überzeugen können, ist Software in dieser Form tatsächlich nur noch durch den Computer zu deuten. Einem Normalsterblichen bleibt verschlossen, was sich darin verbirgt. Eine Ausnahme bildet im gezeigten Beispiel allenfalls der Satz *«This program cannot be run in DOS mode.»*, den man rechts oben erkennen kann.

Es bedarf wohl keiner großen Überzeugungsarbeit, damit Sie den Unterschied zwischen freier (bzw. offener) Software und proprietärer Software anerkennen. Für diejenigen, die doch noch Zweifel haben, hier einige Aussagen aus den zugehörigen Lizenzbestimmungen:<sup>11</sup>

- \* *«Sie sind berechtigt, eine einzige Kopie des SOFTWAREPRODUKTS auf dem COMPUTER zu installieren und zu verwenden.»*
- \* *«Sie sind nicht berechtigt, das SOFTWAREPRODUKT zu vermieten, zu verleasen oder zu verleihen.»*
- \* *«Sie sind nicht berechtigt, das SOFTWAREPRODUKT zurückzuentwickeln, zu dekompileieren oder zu disassemblieren ...»*

Nach diesem Kontrastprogramm möchte ich zum zweiten Spotlight meines Referates kommen, zu der Frage, *wer* denn nun eigentlich solchen Code entwickelt, wie er auf dem ersten Bild zu sehen war. Es geht dabei nicht darum, Biographien einzelner Akteure hinterherzuspüren, vielmehr ist nach einer Typologie gefragt.

## **2. «Who is doing it?»**

**«Who is doing it?»** - unter diesem Titel führten Studenten des Fachbereichs Informatik der TU Berlin im Sommersemester 2001 eine Studie durch (Robles et al. 2001), die zum Ziel hatte,

---

<sup>10</sup> Die Hexadezimaldarstellung ist der binären Darstellung gleichwertig, allerdings -im Druck- kompakter.

<sup>11</sup> Microsoft Windows 98 Zweite Ausgabe Endbenutzer-Lizenzvertrag (EULA).



herauszufinden, was für Leute freie Software entwickeln. Meines Wissens handelt es sich um die weltweit aktuellsten und qualitativ besten Daten, die zur Verfügung stehen.

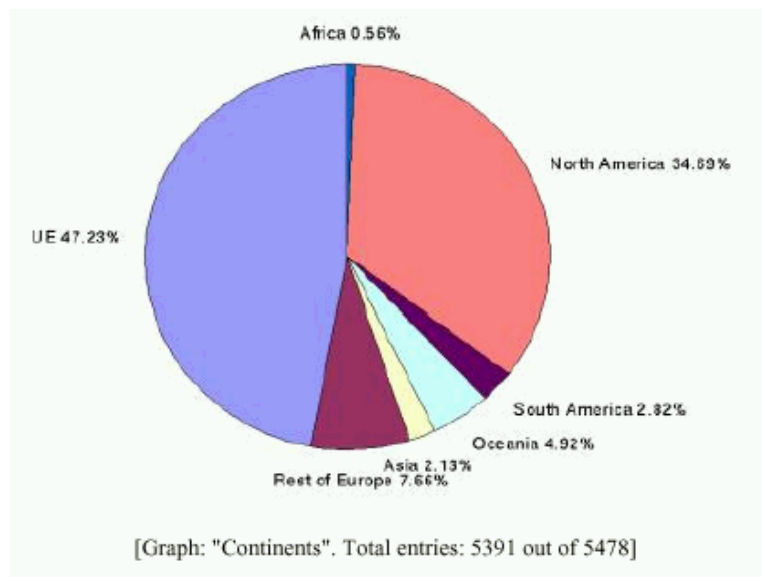
Die Studie ruht auf zwei Säulen:

- 1) Es wurden ca. 1 Gigabyte Quellcode auf Hinweise zu den Urhebern untersucht.  
Um eine Vorstellung davon zu vermitteln: Auf eine herkömmliche CD-ROM passen 650 Megabyte, d.h. 0,65 Gigabyte an Daten.
- 2) Es wurde eine Online-Befragung durchgeführt, an der sich weltweit fast 5600 Entwickler von freier und "Open Source"-Software beteiligt haben. Im Vergleich mit anderen (fachspezifischen) Umfragen ist die Beteiligungsquote außergewöhnlich hoch.

Insbesondere die im Rahmen der Online-Umfrage ermittelten Daten lassen Schlüsse auf die Demographie der Entwickler zu, die mit einigen Vorurteilen zu brechen nahelegen. Ausgewählte Resultate möchte ich kurz vorstellen.

## 2.1 Woher stammen die Entwickler?

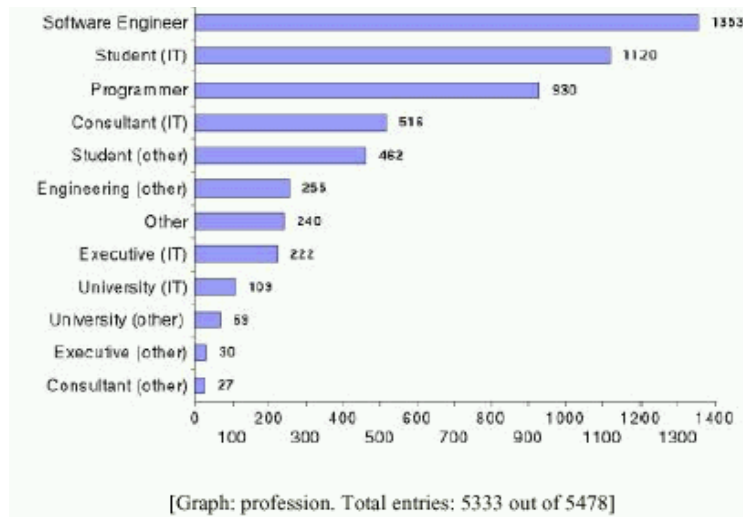
Die folgende Grafik zeigt die Verteilung der Entwickler auf den verschiedenen Kontinenten.



Betrachtet man EU-Staaten und andere europäische Staaten zusammen, so stellt man fest, daß mehr als die Hälfte der Entwickler, nämlich fast 55%, aus Europa stammen. Das Resultat ist erstaunlich, vergleicht man es mit der Situation im Bereich der proprietären Software, wo US-amerikanische Firmen marktbeherrschend sind. Ganz offensichtlich mangelt es Europäern nicht an Kompetenz im

Bereich der Software-Entwicklung. Ihre Kompetenz setzen die europäischen Entwickler -aus zu untersuchenden Gründen- besonders stark für die Entwicklung freier Software ein.

## 2.2 Welche Berufsposition haben die Entwickler?



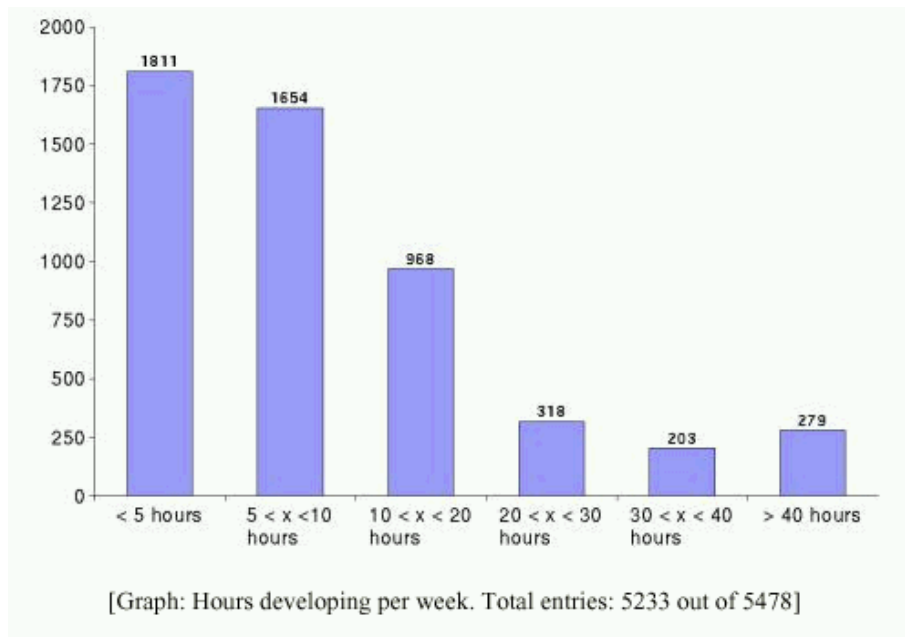
Es dürfte nicht überraschen, daß die große Mehrheit der Entwickler aus dem IT-Bereich stammt, ca. 80%. Im Gegensatz zu verbreiteten Vorurteilen bilden ausgebildete Fachleute den übergroßen Anteil, wenn auch der Beitrag von Studenten nicht zu unterschätzen ist. Das Bild vom unprofessionellen Hacker<sup>12</sup> erweist sich also als Chimäre.

Untersucht man die Arbeitssituation genauer, so stellt die Studie fest, sind etwa zwei Drittel der Entwickler außerhalb des universitären Bereichs angesiedelt (Robles et al. 2001: 32). Die Entwicklung von "freier Software" und "Open Source"-Software ist wesentlich stärker in der Wirtschaft verankert, als gemeinhin vermutet wird. Die Kooperation mit den universitätsnahen Entwicklern -dem restlichen Drittel- dürfte einem raschen Technologietransfer sehr förderlich sein.

---

<sup>12</sup> Vgl. etwa: «*It started as a small rebellion -- A warning shot fired at the Windows monopoly by independent-minded programmers. But the open source movement traditionally associated with the happy penguin and the pierced, tattooed crowd is increasingly moving into the enterprise, mingling peacefully with commercial and proprietary code.*» (Babcock 2001)

## 2.3 Zeitaufwand für die Software-Entwicklung



Zwei Drittel der Entwickler wenden weniger als 10 Stunden pro Woche auf, um "freie Software" zu entwickeln.

Diese Zahl kann verblüffen. Wie kann es gelingen, in weniger als 10 Stunden pro Woche nennenswerte Anteile an Betriebssystemen, Office-Paketen, Webservern usw. zu entwickeln? Nur ein Schluß erscheint zulässig: Ganz offensichtlich haben die Entwickler eine sehr effiziente Form der Arbeitsteilung für die Software-Herstellung gefunden. Daß die Offenheit der Lizenzbedingungen dabei eine wesentliche Rolle spielen dürfte, kann gemutmaßt werden.

Die Studie hat eine Reihe weiterer, sehr interessanter Resultate erbracht. Wer sich dafür interessiert, sei auf das Internet verwiesen, wo die Studie heruntergeladen werden kann.

## 3. Zur Motivation

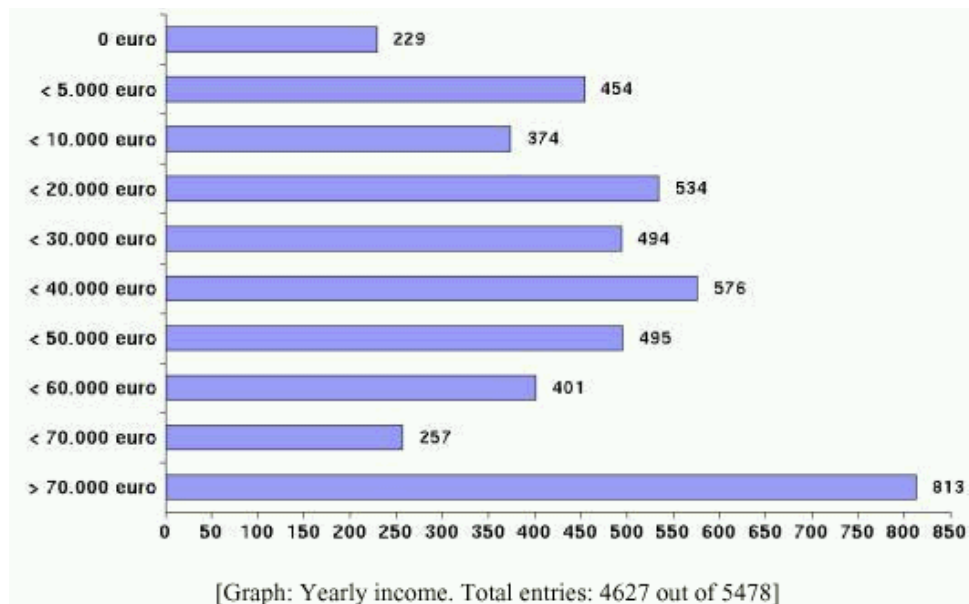
Ich möchte mich nun dem dritten Spotlight meines Vortrages zuwenden und nach der *Motivation* der Entwickler fragen: "*Warum tun die das?*"

Hintergründe von Motivation auszuleuchten ist alles andere als trivial, zumal wenn es um die Realität geht. Es ist nicht wie unter Laborbedingungen, es lassen sich keine festen Grenzen ziehen und man bewegt sich unweigerlich auf schwankendem Grund. Einige der in der Studie erhobenen Daten

lassen Indizien ausmachen, die uns dem Thema der Diskussion um das "virtuelle Eigentum" näher bringen können.

### 3.1 Die wirtschaftliche Lage der Entwickler?

Betrachten wir zuerst die wirtschaftliche Situation, in der sich die Entwickler befinden. Kriterium dafür ist primär das Jahreseinkommen.

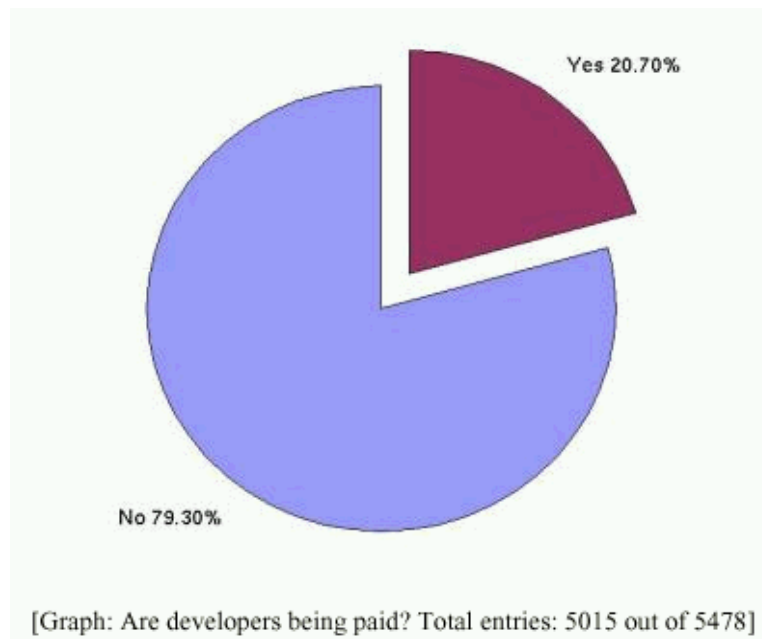


Wie man sehen kann, stammen die Entwickler aus allen Einkommensklassen und tendenziell verdienen sie nicht schlecht.

Man erkennt einen Schwerpunkt im Bereich verhältnismäßig hoher Einkommen, von über 70.000 Euro im Jahr. Ob dieser Wert realistisch ist, oder ob -wie die Autoren der Studie vermuten- die Befragten übertrieben haben, läßt sich ohne weiteres nicht feststellen. Da indessen hochqualifizierte Software-Entwickler in den USA wesentlich besser bezahlt werden als in Europa, könnten sie diese Gehaltsstufe ausfüllen. Aber das bleibt letztlich Spekulation und könnte nur durch weitere Untersuchungen aufgeklärt werden.

### 3.2 Bezahlung für freie Software-Entwicklung?

Interessant ist es in meinen Augen, wie es mit der Bezahlung im Hinblick auf die Entwicklung freier Software aussieht. Auch danach wurde in der Studie gefragt. Das Ergebnis fiel eindeutig aus.



Vier Fünftel der Entwickler werden für ihre Arbeit an freier Software nicht bezahlt. Lediglich bei 20% handelt es sich um Erwerbsarbeit. An diesem Befund gibt es nichts zu bezweifeln. Jedoch erklärt auch er nicht, warum die Software-Entwickler freiwillig freie Software entwickeln und kostenlos<sup>13</sup> weitergeben.

### 3.3 Unzufriedenheit im Beruf?

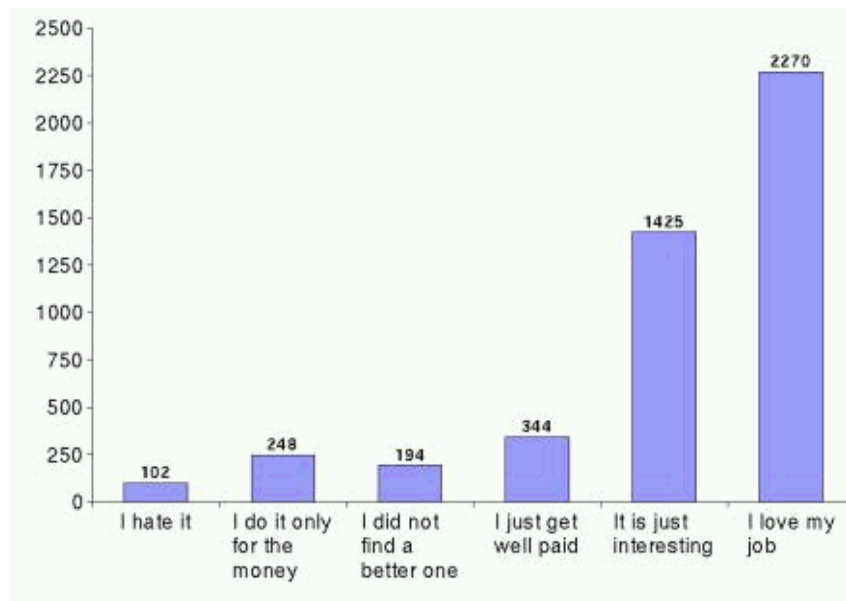
Man könnte nun auf die Idee verfallen, daß es sich dabei quasi um ein Hobby handelt, mit dem der Frustration im Beruf entgegengearbeitet wird.

Daß die Zufriedenheit mit der beruflichen Tätigkeit in den Industrieländern seit Jahrzehnten abnimmt, ist durch eine Vielzahl von Untersuchungen belegt.<sup>14</sup> Wie sieht es damit bei den Entwicklern freier Software aus?

---

<sup>13</sup> Um das 'kostenlos' begrifflich präziser zu fassen, müßte man eigentlich von 'lizenzgebührenfrei' sprechen. Mit der Erlangung der Software können tatsächlich Kosten verbunden sein: Telefonkosten für den Download, Kosten für CD-Rohlinge usw.

<sup>14</sup> Vgl. z.B. (Lane 1999).



Laut Studie sind ca. 80% mit ihrer beruflichen Tätigkeit -und damit im weltweiten Maßstab überdurchschnittlich viele- zufrieden<sup>15</sup>: Sie finden ihre Arbeit interessant oder erklären sogar, sie zu "lieben".

Mein Fazit lautet: Frustration mit der Marktwirtschaft als solcher ist nicht zu erkennen. Im Gegenteil scheinen sowohl die Einkommensverhältnisse (3.1) als auch die Zufriedenheit im Beruf (3.2) eher darauf hinzuweisen, daß die Entwickler sich nicht im Gegensatz zur marktwirtschaftlichen Ordnung sehen. Ihnen daher sozialistische Anwandlungen anzudichten,<sup>16</sup> oder naiven Altruismus zu unterstellen,<sup>17</sup> erscheint mir unangemessen.

### 3.4 Was sagt die Ökonomik?

Die Ökonomik tut sich schwer mit Erklärungen.

---

<sup>15</sup> Ich interpretiere hier positive Äußerungen zur beruflichen Situation im Sinne von Zufriedenheit. Über die Berechtigung einer solchen Auslegung ließe sich natürlich diskutieren.

<sup>16</sup> Darauf richtete sich etwa auch die Fragestellung für diese Diskussion. Eine lesenswerte Auseinandersetzung mit der Frage nach einer etwaigen sozialistischen Attitüde ist (Prasad 2001).

<sup>17</sup> So etwa (Tauchert 2000).

Eine Reihe von Autoren<sup>18</sup> hat zwar nachgewiesen, daß Unternehmen letzten Endes von den Ergebnissen des Entwicklungsprozesses (in Form einsatzfähiger, kostengünstiger, wartungsarmer Software) profitieren: Sie sparen Entwicklungskosten und Lizenzgebühren. Befriedigende ökonomische Begründungen für die andere Seite der Medaille, für das Verhalten der Entwickler, sind jedoch nicht recht auszumachen.<sup>19</sup>

Ein Ursache dafür könnte darin liegen, daß den Betrachtungen ein unpassendes Modell zugrundegelegt wird.

Da es sein kann, daß nicht jeder der Anwesenden eine Vorstellung davon hat, mit welchem Menschenbild die Wirtschaftswissenschaften operieren, sollte ich dessen wesentliche Merkmale kurz ansprechen.

Ausgangspunkt ist -im Sinne der herrschenden Meinung- der '*homo oeconomicus*'. Folgende Definition dafür kann man einem ökonomischen Lehrbuch entnehmen (Homann/Suchanek 2000: 58):

*«Individuen maximieren ihren Nutzen unter Restriktionen.»<sup>20</sup>*

Umgangssprachlich ausgedrückt,<sup>21</sup> besagt das Modell des *homo oeconomicus*, daß (1) Individuen -und nicht etwa Gruppen von Menschen- Ausgangspunkt der Erklärung sind, (2) das Individuum seine Entscheidungen rational in Anbetracht einer überschaubaren Reihe von Anreizen<sup>22</sup> trifft und (3) diese Anreize von außen gesetzt werden. Ziel der Entscheidungen ist die Maximierung des Eigennutzes bzw. -in etwas abgeschwächter Form-<sup>23</sup> die *Satisfizierung* der eigenen Bedürfnisse.

---

<sup>18</sup> Vgl. z.B. (Lerner/Tirole 2000; Weber 2000; Nüttgens/Tesei 2000a; Nüttgens/Tesei 2000b; Nüttgens/Tesei 2000c; Ardal 2000)

<sup>19</sup> Ansätze finden sich etwa bei (Raymond 1999).

<sup>20</sup> Den terminus technicus *Restriktionen* kann man auch mit *Anreizen* übersetzen (ebda).

<sup>21</sup> Vgl. (Fritsch/Wein/Ewers 2001: 30f).

<sup>22</sup> *Anreize* umfaßt dabei sowohl positive Anreize (Verantwortung, Belohnung etc.) wie negative Anreize (Beschränkungen, Strafen etc.).

<sup>23</sup> Vgl. etwa (Simon 1994)

Ausgehend von solchen Modellannahmen erscheint es nur logisch, wenn die Notwendigkeit geistigen Eigentums im Bereich des Urheber- und Patentrechtes mit seiner *Anreizfunktion* begründet wird. In allen Bereichen, in denen es um die Rechtfertigung von 'intellectual property' geht, stoßen wir auf Seiten der Befürworter auf Formulierungen wie diese:

«Wenn Urheber und ausübende Künstler weiter schöpferisch und künstlerisch tätig sein sollen, müssen sie für die Nutzung ihrer Werke eine angemessene Vergütung erhalten, was ebenso für die Produzenten gilt, damit diese die Werke finanzieren können.»<sup>24</sup>

Die Botschaft ist klar: Ohne Vergütungsansprüche gibt es keine Investitionen in kreative Tätigkeiten. Die Vergütungsansprüche sollen Bedürfnisse wecken und der Weg der Bedürfnisbefriedigung soll über die Kreation geistiger Produkte verlaufen. Das verkäufliche geistige Eigentum stellt Gewinne in Aussicht, Gewinne als Belohnung für zu vollbringende Anstrengungen.<sup>25</sup> So weit, so gut.

Versucht man jetzt aber, mit demselben Modell die Entwicklung freier Software zu ergründen, wird es hoch problematisch. Die Anreizfunktion in Aussicht gestellter Gewinne ist praktisch<sup>26</sup> nicht mehr zu erkennen. Statt sich zu Unternehmen zusammenzuschließen und Software zu verkaufen, um Gewinne zu erzielen, investieren die Entwickler Zeit und Kraft, zum Teil auch Geld, und verzichten auf den Eigentumsanspruch. Statt Software zu vermarkten, wird sie -auf der Distributionsebene- dem Markt weitgehend entzogen. Statt die kostenlose Verbreitung von Code mit allen Mitteln zu bekämpfen (wie etwa die in der *Business Software Alliance* zusammengeschlossenen Unternehmen), beseitigen die Entwickler Hindernisse auf dem Weg zur Verbreitung ihrer Software.

---

<sup>24</sup> Aus Punkt (10) der Begründung der **Richtlinie 2001/29/EG** v. 22. Mai 2001 zur Harmonisierung bestimmter Aspekte des Urheberrechts und der verwandten Schutzrechte in der Informationsgesellschaft (Urheberrechts-Richtlinie).

<sup>25</sup> Vgl. (Litman 2001: 101ff).

<sup>26</sup> Ausnahmen bilden vielleicht die Unternehmen, die mit "Open Source"-Produkten bzw. der Vermarktung von freier Software und entsprechenden Dienstleistungen Gewinne erzielen wollen. Deren nicht unbedingt positive wirtschaftliche Lage spricht nicht gegen die hier vorgebrachte These, sondern untermauert sie eher noch.



### 3.5 Zusammenfassung

Sieht man die Resultate der Studie zusammen, so hat es den Anschein, als gäbe es auf Seiten der Entwickler keinen ökonomischen Anlaß, freie Software zu entwickeln.<sup>27</sup> Jedenfalls nicht im Sinne des ‘*homo oeconomicus*’-Modells der Wirtschaftswissenschaften. Das Eigentum<sup>28</sup>, könnte man sagen, hat -in diesem Bereich der Software-Entwicklung- ausgedient, weil es sich als *ein ineffizientes Instrument* erwiesen hat.<sup>29</sup> Sich bei der Behandlung des Themas auf Eigentum<sup>30</sup> zu fixieren, führt dann in die Irre.

### 4 ‘homo faber’ versus ‘homo oeconomicus’ (!?)

Diesen kleinen empirischen Exkurs möchte ich mit einer Provokation beenden.

Ich möchte einen Perspektivwechsel vorschlagen.

Ansatzpunkte dazu liefern etwa Forschungsergebnisse von Ökonomen aus der Schweiz und Psychologen aus den USA (vgl. Frey 1999; Frey/Jegen 2000; Deci/Ryan/Koestner 1999a; Deci/Ryan/Koestner 1999b; Osterloh/Frey 1999; Frey/Götte 1999). Diese haben untersucht,

---

<sup>27</sup> Sehr wohl aber gibt es Anreize, freie Software zu nutzen.

<sup>28</sup> Die Begrifflichkeit von Eigentum verstehe ich hier in folgendem Sinne: «*Das (privatrechliche) E. als das grundsätzlich unbeschränkte Herrschaftsrecht über eine Sache berechtigt den Eigentümer regelmäßig, mit der Sache nach Belieben zu verfahren [...] und andere (Unberechtigte) von jeder Einwirkung auszuschließen [...]*» (Creifelds 1996: 336) bzw. «*Eigentum, nach versch. sozialphilos. Lehren ein überpositives, d.h. jegl. menschl. Rechtsetzung vor- u. übergeordnetes Recht des Menschen. Rechtl. die Verfügungsgewalt über Sachen bzw. Güter der Umwelt des Menschen.*» (Hillmann 1994: 170) bzw. -in erweitertem Sinne- «*Property rights [...] This requires that the exercise of rights not be interfered with by others. Uses of resources not legitimated by the user's possession of property rights are illegal by definition [...]*» (Demsetz 1998: 144)

<sup>29</sup> Spezifische Anforderungen an die Sicherheit von Software etwa, lassen sich möglicherweise überhaupt nur dann erfüllen, wenn keine Eigentumsansprüche gestellt werden, sondern stattdessen dem Prinzip der Nicht--Ausschließlichkeit gefolgt wird. Vgl. (Gehring 2001).

<sup>30</sup> Auch *Access statt Eigentum* im Sinne von *Jeremy Rifkin* ändert daran nichts. Im Kern geht es (in beiden Fällen) um die exklusive Kontrolle zum Zwecke der Vermarktung, wie im Grunde auch (Rifkin 2000: 12) konstatiert: «*Im Zeitalter der Netzwerke gewinnen Anbieter, die wertvolles geistiges Kapital aufgehäuft haben, zunehmend Macht über die Bedingungen, unter denen Nutzer auf gewinnträchtige Ideen, Wissen und Fachkenntnisse zugreifen können.*» Und eben diese exklusive Kontrolle über die Bedingungen des Zugriffs stellt sich als das primäre Hindernis auf dem Weg zu mehr Effizienz bei der Software-Entwicklung dar.

wodurch Menschen zum Handeln motiviert werden. Ihre Forschungen haben gezeigt, daß insbesondere im Bereich der kreativen Tätigkeiten die Eigenmotivation, die sogenannte *intrinsische Motivation* der Akteure einen wesentlichen Einfluß hat. Demgegenüber, so hat es sich gezeigt, sind äußere Anreize -sogenannte *extrinsische Faktoren*- zum Teil sogar schädlich (Frey 1999; Frey/Jegen 2000).<sup>31</sup> Belohnungen im Sinne von 'intellectual property rights' könnten dazu gehören.

**Wir sollten uns in der Betrachtung der Entwicklungen im Internet nicht immer nur auf ausgetretenen Pfaden bewegen. Vielleicht ist es an der Zeit, unsere Modellannahmen über die Handlungsmotive von Menschen zu hinterfragen: Statt immer den 'homo oeconomicus' zu suchen, wo er nicht zu finden ist,<sup>32</sup> könnten wir den 'homo faber' entdecken.**

---

<sup>31</sup> Nachweislich können sich Belohnungen dort kontraproduktiv auswirken, wo die Akteure voluntaristisch tätig werden (wie es bei kreativen Tätigkeiten regelmäßig der Fall ist). Statt mehr zu "leisten", sinkt die Leistung, wenn Belohnung in Aussicht gestellt wird. Die Ökonomen sprechen dann von einem Verdrängungseffekt (crowding-out effect). Vgl. etwa (Frey/Jegen 2000: S.17): «*Intrinsic motivation has been argued to be important when it comes to volunteering (Freeman, 1997). The authors use a unique data set from Switzerland to evaluate how financial rewards to volunteers affect their intrinsic motivation. The incidence of rewards is found to reduce the amount of volunteering.*»

<sup>32</sup> Auch andere Forschungen weisen darauf hin, daß das Modell des 'homo oeconomicus' revisionsbedürftig zu sein scheint. Vgl. z.B. (Falk 2001)

## Referenzen

Ardal, Atila (2000)

**Open Source - das Beispiel Linux. Ökonomische Analyse und Entwicklungsmodell eines erfolgreichen Betriebssystems.** Diplomarbeit, 2000, <http://ig.cs.tu-berlin.de/da/059/index.html> [14.11.2001]

Babcock, Charles (2001)

**Forced Fit.** Interactive Week, May 14, 2001, im Internet: <http://www.interactiveweek.com/article/0,3658,s%253D619%2526a%253D11850,-00.asp> [7. Nov. 2001]

Creifelds (1996)

**Rechtswörterbuch.** 13. Aufl., Verlag C.H. Beck, München, 1996.

Deci, E.L.; Ryan, R.M.; Koestner, R. (1999a)

**A Meta-Analytic Review of Experiments Examining the Effects of Extrinsic Rewards on Intrinsic Motivation,** Psychological Bulletin, 125 (1999), pp 627-668.

Deci, E.L.; Ryan, R.M.; Koestner, R. (1999b)

**The Undermining Effect Is a Reality After All - Extrinsic Rewards, Task Interest, and Self-Determination,** Psychological Bulletin, 125 (1999), pp 692-700.

Demsetz, Harold (1998)

**Property Rights.** pp 144-155, in: Peter G. Newman (ed.): The New Palgrave Dictionary of Economics and the Law, Vol. 3, Macmillan Reference Limited, London, 1998.

Falk, Armin (2001)

**Homo Oeconomicus Versus Homo Reciprocans: Ansätze für ein Neues Wirtschaftspolitisches Leitbild?**, Working Paper No. 79, July 2001, Institute for Empirical Research in Economics, University of Zurich, online: <http://www.unizh.ch/iew/wp/iewwwp079.pdf> [18.11.2001]

Frey, Bruno S. (1999)

**Institutions and morale: the crowding-out effect.** pp 437-460, in: Avner Ben-Ner, Louis Putterman (eds.): Economics, Values, and Organization, Cambridge University Press, Cambridge, 1999.

Frey, Bruno S.; Götte, Lorenz (1999)

**Does Pay Motivate Volunteers?**, Working Paper No. 7, May 1999, Institute for Empirical Research in Economics, University of Zurich, online: <http://www.unizh.ch/iew/wp/iewwwp007.pdf> [18.11.2001]

Frey, Bruno S.; Jegen, Reto (2000)

**Motivation Crowding Theory: A Survey of Empirical Evidence.** Revised Version, Working Paper No. 49, Juni 2000, Institute for Empirical Research in Economics, University of Zurich, online: <http://www.unizh.ch/iew/wp/iewwwp049.pdf> [18.11.2001]

Michael Fritsch, Thomas Wein, Hans-Jürgen Ewers (2001)

**Marktversagen und Wirtschaftspolitik.** 4. Aufl., Verlag Franz Vahlen, München, 2001

Gehring, Robert A. (2001)

**"Software Patents" - IT-Security at Stake?** Paper presented at the congress "Innovations for an e-Society. Challenges for Technology Assessment", Oct 17-19, 2001, Berlin, Germany, <http://ig.cs.tu-berlin.de/ap/rq/index.html> [14.11.2001]

- Hegel, Georg Wilhelm Friedrich (1821)  
**Grundlinien der Philosophie des Rechts.** 4. Aufl., Johannes Hofmeister (Hrsg.): G.W.F. Hegel, Sämtliche Werke, Band XII, Philosophische Bibliothek Band 124a, Felix Meiner Verlag, Hamburg, 1955.
- Hillmann, Karl-Heinz (1994)  
**Wörterbuch der Soziologie.** 4. Aufl., Alfred Kröner Verlag, Stuttgart, 1994.
- Homann, Karl; Suchanek, Andreas (2000)  
**Ökonomik. Eine Einführung.** Mohr Siebeck, Tübingen, 2000.
- Lane, Robert E. (1999)  
**The joyless market economy.** pp 461-488 in Avner Ben-Ner, Louis Putterman (eds.): Economics, Values, and Organization, Cambridge University Press, Cambridge, 1999.
- Lerner, Josh; Tirole, Jean (2000)  
**The Simple Economics of Open Source.** February 25, 2000,  
<http://papers.nber.org/papers/W7600> [3.12.2000]
- Litman, Jessica (2001)  
**Digital Copyright.** Prometheus Books, Amherst, NY, 2001.
- Nüttgens, Markus; Tesei, Enrico (2000a)  
**Open Source: Konzept, Communities und Institutionen.** IWi-Heft 156, <http://www.iwi.uni-sb.de/iwi-hefte/heft156.pdf> [14.11.2000]
- Nüttgens, Markus; Tesei, Enrico (2000b)  
**Open Source: Produktion, Organisation und Lizenzen.** IWi-Heft 157, <http://www.iwi.uni-sb.de/iwi-hefte/heft157.pdf> [14.11.2000]
- Nüttgens, Markus; Tesei, Enrico (2000c)  
**Open Source: Marktmodelle und Netzwerke.** IWi-Heft 158, <http://www.iwi.uni-sb.de/iwi-hefte/heft158.pdf> [14.11.2000]
- Osterloh, Margit; Frey, Bruno S. (1999)  
**Motivation, Knowledge Transfer, and Organizational Form,** Working Paper No. 27, Nov. 1999, Institute for Empirical Research in Economics, University of Zurich, online:  
<http://www.unizh.ch/iew/wp/iewwwp027.pdf> [18.11.2001]
- Prasad, Ganesh (2001)  
**The Capitalist View of Open Source.** LinuxToday, 18. Mai 2001,  
[http://linuxtoday.com/news\\_story.php3?ltsn=2001-05-18-007-20-OP](http://linuxtoday.com/news_story.php3?ltsn=2001-05-18-007-20-OP) [14.11.2001]
- Raymond, Eric S. (1999)  
**The Cathedral and the Bazaar.** O'Reilly, Sebastopol, CA, 1999.
- Rifkin, Jeremy (2000)  
**Access. Das Verschwinden des Eigentums,** Campus Verlag, Frankfurt/New York, 2000.
- Richtlinie 2001/29/EG**  
v. 22. Mai 2001 zur Harmonisierung bestimmter Aspekte des Urheberrechts und der verwandten Schutzrechte in der Informationsgesellschaft
- Robles, Gregorio; Scheider, Hendrik; Tretkowski, Ingo; Weber, Niels (2001)  
**Who Is Doing It? A research on Libre Software developers.** Fachgebiet Informatik und Gesellschaft, TU-Berlin, August 2001, <http://iq.cs.tu-berlin.de/s2001/ir2/ergebnisse/OSE-study.html> [2.11.2001]

Simon, Herbert A. (1994)

**Die Wissenschaft vom Künstlichen.** Springer-Verlag, Wien, New York, 1994.

Tauchert, Wolfgang (2000)

**Wem gehört das Wissen? Geistiges Eigentum im Zeitalter des Internet.** Positionspapier für die Tagung, veranstaltet von der Heinrich-Böll-Stiftung am 20./21.10.2000 in Berlin.

Weber, Steven (2000)

**The Political Economy of Open Source Software.** BRIE Working Paper #140, 2000,  
<http://brie.berkeley.edu/~briewww/pubs/wp/wp140.pdf> [14.11.2001]