

Freiheit und Software

Georg Jakob

*Abteilung für Rechtsinformatik am IGR, Universität Salzburg
Churfürststrasse 1, A-5020 Salzburg
jack@unix.sbg.ac.at*

Schlagworte: Lizenz, Lizenzvertrag, Urheberrecht, Computervertrag, Softwarevertrag, Softwareerwerb, Schenkung, Gesellschaft Bürgerlichen Rechts, Freie Software, Free Software, FSF, General Public License, GNU, Linux, Open Source

Abstract: Dieser Beitrag nähert sich dem Phänomen freier Softwarelizenzen unter verschiedenen Gesichtspunkten: Zunächst wird versucht zu zeigen, warum es überhaupt zur Proprietarisierung, dh der Verwendung der heute üblichen, restriktiven Lizenzen und dem unter Verschlusshalten von Quellcodes kam. Dann wird die General Public License, also jene Lizenz unter der auch der Linux-Kernel vertrieben wird einer juristischen Beurteilung unterzogen. Abschließend werden Erklärungsansätze dafür gesucht, warum gerade die GPL sowohl aus wirtschaftlichen als auch rechtlichen Gründen attraktiv sein kann.

„Omnis enim res, quae dando non deficit, dum habetur et non datur, nondum habetur, quomodo habenda est.“

Augustinus, De doctrina christiana

1. Die Quelle der Freiheit

Wer verzweifelt nicht gelegentlich an abstürzenden Programmen, hängengebliebenen Druckaufträgen oder anderen Problemen einer Technologie, deren Idee es eigentlich ist, unser Leben zu *erleichtern*? Heute werden Computersysteme hauptsächlich durch das Klicken bunter Schaltflächen administriert. Wo dies nicht mehr weiterhilft, kann nur noch auf den Herstellersupport oder die nächste, „verbesserte“ Programmversion gewartet werden. Doch dem war nicht immer so – denn am Anfang war der Quellcode¹ frei. Zugänglicher Quellcode eröffnet die Möglichkeit – die *Freiheit* – ein Problem selbst zu lösen, oder die entsprechende Lösung

¹ Der Quellcode eines Programmes wird als ASCII-Text in einer Programmiersprache (zB C) erstellt. Mittels eines Compilers wird daraus eine Binärdatei, also das ausführbare Programm generiert.

eines anderen zu verwenden und so dem „Warten auf Godot“ ein Ende zu setzen.

2. Proprietarisierung

Noch bis Mitte der 70er Jahre wurden Computerprogramme fast ausschließlich im Quellcode verbreitet. Ein ganzes Betriebssystem – Unix², wurde so an den AT&T Bell Labs ent- und von Universitäten in aller Welt weiterentwickelt. Mit ihm erblickte auch die Programmiersprache C das Licht der Welt, womit die Erstellung portabler Programme wesentlich erleichtert wurde. Als schließlich mit dem Altair der erste PC auf den Markt kam, wurde auf dem neu entstandenen Massenmarkt zunächst ebenso frei mit dem Code umgegangen wie bisher.³ Einer der ersten, die hier neue Verdienstmöglichkeiten sahen, war *William Gates III*, der in seinem „Open Letter to Hobbyists“⁴ die gängige Praxis der Programmweitergabe als Diebstahl anprangerte und auf mehr und mehr Unterstützung seitens der Industrie bauen konnte. Und tatsächlich wurde in den kommenden Jahren die Weitergabe von Programmen einerseits lizenzrechtlich, andererseits auch faktisch sehr effektiv⁵ eingeschränkt: Software wurde nur mehr im bloß maschinenlesbaren Binärcode vertrieben. Dass die Hersteller versuchten, durch das Nichtoffenlegen Ihrer Standards Kunden an sich zu binden, durfte auch die Entstehung des bekannten Marktmonopols auf dem Sektor der Betriebssystemsoftware begünstigt haben. Nur so konnte ein System wie DOS, das ursprünglich für simple Hardwaretests konzipiert war⁶, zum Standard internationaler Datenverarbeitung werden. Unix, dessen Copyright bei AT&T, dem Arbeitgeber seiner Erfinder lag, wurde zu einer Betriebssystemfamilie, deren zahlreiche Mitglieder miteinander inkompatibel wurden. Anfangs schienen nur Pro-

² Eine interessante Darstellung der Geschichte von Unix stammt von *Moffit*, zu finden unter: <http://crackmonkey.org/unix.html>; Vgl auch *Rosenberg*, Open Source – The Unauthorized White Papers, Foster City/CA (2000), S 3; Sowie *Raymond*, The Cathedral and the Bazaar, Sebastopol/CA (1999), S 7. Letzteres ist online unter <http://tuxedo.org/~esr/writings/> Alle zitierten Links wurden am 23.5. 2001 überprüft und erwiesen sich als funktionierend, weshalb im weiteren auf die Angabe eines Datums verzichtet wird.

³ *Rosenberg*, Open Source, S 9.

⁴ Zu finden unter: <http://www.blinkenlights.com/classiccmp/gateswhine.html>.

⁵ Dies wurde später durch entsprechende Urheberrechtsnovellen weder gefördert (BGBl 93/1999).

⁶ *Tanenbaum*, Modern Operating Systems, Upper Saddle River/NJ (1992), S 316.

grammierer, wie *Richard Stallman* vom MIT Artificial Intelligence Lab, unter diesen Zuständen zu leiden, da sie nun Fehler in Programmen nicht mehr selbstständig beheben, geschweige denn Software konstruktiv weiterentwickeln konnten. Und wer den Quellcode eines Programms ausnahmsweise zu sehen bekam, mußte ein Non Disclosure Agreement unterzeichnen, das ihm jede Weitergabe seines Wissens untersagte.⁷ Der effizienteste Weg, ein programmiertechnisches Ziel zu erreichen,⁸ war damit versperrt.

3. Die Free Software Foundation

Stallman gründete 1984 die Free Software Foundation⁹ und rief deren wichtigstes Projekt ins Leben: Die Schaffung, eines Unix-ähnlichen Betriebssystems, das für jedermann im Quellcode verfügbar sein sollte. Im Zentrum seiner Überlegungen stand dabei von Anfang an nicht die Kostenfreiheit, sondern die Freiheit des Wissenstransfers: *Think of free speech, not free beer*. Da AT&T noch immer die Schutzrechte am Markennamen Unix hielt, nannte er das Projekt GNU¹⁰ – ein rekursives Akronym für „GNU's not Unix“. Die wichtigsten Beiträge zu GNU kamen zunächst von ihm selbst: Der Compiler GCC, der Editor Emacs und die Lizenz unter der ganze System stehen sollte: Die General Public License, die GPL¹¹. Obwohl beispielsweise GCC aufgrund seiner Leistungsfähigkeit in Fachkreisen Begeisterung auslöste fehlt dem GNU-System bis heute der zentrale Baustein¹²: Ein stabiler Betriebssystemkern, ein Kernel¹³. Nachdem *Linus Torvalds*, ein Student der Computerwissenschaften aus Finn-

⁷ *Stallman*, The GNU Project, <http://www.gnu.org/gnu/thegnuproject.html>

⁸ *Ravicher*, Facilitating Collaborative Software Development, Virginia Journal of Law and Technology, <http://www.vjolt.net/vol5/issue3/v5i3a11-Ravicher.html>.

⁹ <http://www.fsf.org>.

¹⁰ <http://www.gnu.org>.

¹¹ <http://www.fsf.org/copyleft/gpl.html>.

¹² Es fehlte keineswegs am Versuch: Der GNU HURD soll als Microkernel die Hohe Schule der Computerwissenschaft in die Praxis umzusetzen und erschien erstmals 2000 (also nach fast 10 Jahren) in einer ansatzweise benutzbaren Version 0.3, Vgl <http://www.gnu.org/software/hurd/hurd.html>.

¹³ Der Kernel ist sozusagen das Herzstück eines Betriebssystems, der anderen Programmen (auch der Kommandozeile oder einer graphischen Oberfläche) das Ansprechen der vorhandenen Hardware überhaupt erst ermöglicht. Ihm obliegt die Verwaltung dieser Ressourcen und aller im Arbeitsspeicher laufenden Programme als Prozesse.

land erfolglos ein mit Unix vergleichbares Betriebssystem für seinen 386er PC gesucht hatte, schickte er sich 1990 an, diese Lücke zu schließen, indem er eine der ersten Versionen seines selbstgeschriebenen Kerns unter die GPL und ins Internet stellte und um Mithilfe bat. Damit trat er eine Lawine los: Das Internet ermöglichte es Programmieren weltweit, sich zu beteiligen und so aus dem Hobby eines Erstsemestrigen eine ernstzunehmende Konkurrenz¹⁴ für den größten Softwarekonzern der Welt zu machen.

4. Freie Software contra Open Source?

Torvalds war es auch, der die verteilte Entwicklungsstrategie des Internets perfektionierte, indem er zeigte, dass eine große Zahl von Programmieren unter chaotisch scheinenden Umständen, nur über das Internet verbunden, wesentlich effektiver arbeiten kann als eine von einem Unternehmen bezahlte, hierarchisch organisierte Projektgruppe. Dem traditionellen Modell des Cathedral Building stellte er den Bazar gegenüber¹⁵. Er benutzte den *GCC*, um seinen Kernel, den er *Linux*¹⁶ nannte, zu kompilieren und zusammen mit den anderen, vom GNU-Projekt bereitgestellten Programmen und Tools ergab sich tatsächlich ein komplettes, frei verfügbares Betriebssystem: GNU/Linux. Doch weder GNU noch Linux stehen alleine. *Alle* wichtigen Serverdienste des Internet basieren auf freier Software, etwa *BIND* für den Domain Name Service oder *Sendmail* für eMail. Und ohne den Webserver *Apache* und die Scriptsprache *Perl* hätte es auch den eBusiness Boom nie gegeben¹⁷. Als *Netscape* 1998 den Quellcode des Communicator offenlegte, wurde erstmals eine breitere Öffentlichkeit auf das Phänomen der freien Software aufmerksam. Allerdings unter einem anderen Namen: Open Source. Die neugegründete Open Source Initiative (OSI)¹⁸ betont im Gegensatz zur FSF nicht den Aspekt der Freiheit, sondern die technische Überlegenheit und die wirtschaftlichen Vorteile zugänglichen Quellcodes und versucht bewußt, sich von der FSF abzuheben, der oft eine zu starke Ideologisierung, welche zwangsläufig zu einem Außenseiterdasein freier Software führen müsse,

¹⁴ Linux Grabs Market Share, Wall Street Journal Europe, 10. 4. 2001, S 1 u 25.

¹⁵ Zur Effektivität dieser Arbeitsweise *Raymond*, The Cathedral and the Bazaar, S 27.

¹⁶ Allgemeine Informationen zu Linux gibt es unter: <http://www.linux.org> Die Kernelquellen können von <http://www.kernel.org> heruntergeladen werden.

¹⁷ *Moody*, Die Software Rebellen, Landsberg am Lech (2001), 255. Zur führenden Position von Apache <http://www.netcraft.com/survey>.

¹⁸ <http://www.opensource.org>.

vorgeworfen wurde. Zwar stehen im Zentrum der Open Source Definition wiederum die GPL und die Free Software Guidelines des nichtkommerziellen (und der FSF nahestehenden) GNU/Linux Distributors Debian¹⁹, doch anerkennt und befürwortet sie neben der GPL auch Lizenzen, die dem Lizenzgeber bedeutende Privilegien einräumen, wie etwa die Netscape Public License oder die Apple Public Source License des neuen OS-X Kernels. Dies ist auch insofern problematisch, als die GPL das Linken von freiem Code mit nicht freiem grundsätzlich untersagt²⁰. Werden nun Programme, die zwar mit offenem Quellcode vorliegen, aber anderen Einschränkungen unterliegen (etwa jederzeitige Widerrufbarkeit, Übergang des Copyrights an einem Bugfix auf den Hersteller eines Programms) mit freier Software verbunden, führt dies einerseits zu Rechtsunsicherheit hinsichtlich der Kompatibilität der unterschiedlichen Lizenzen und andererseits zur schon bekannten Gefahr der Repropriarisierung²¹.

5. Die General Public License

Das Hauptziel bei der Gründung der FSF war die Bereitstellung eines kompletten Betriebssystems mit freiem Zugang zum gesamten Quellcode. Der auf den ersten Blick logische Weg dazu wäre, die entsprechende Software als Public Domain frei- und damit den Urheberrechtsschutz nahezu aufzugeben. Doch da die Vergangenheit gezeigt hatte, dass einmal offener Quellcode keineswegs dauerhaft frei bleiben mußte, sollte der Weg diesmal über das Urheberrecht führen. Dessen Grundidee besteht traditionell darin, die Rechte des Urhebers zu sichern, indem jene des Nutzers eingeschränkt werden.²² Die GPL dreht nun genau dieses Konzept um: Als Lizenz stellt sie zunächst klar, dass der Urheber sehr wohl den Anspruch an seiner Schöpfung behält. Doch dann wird genau dieses

¹⁹ <http://www.debian.org>.

²⁰ Zwar erlaubt dies die alternativ verwendbare Lesser GPL

(<http://www.fsf.org/copyleft/lesser.html>) unter bestimmten Voraussetzungen, doch besteht ein weiteres Problem: Die GPL und die LGPL erlauben *alles, außer* seinem Nächsten weniger Rechte zuzugestehen, als man selbst an einem Programm/einer Library hat.

²¹ FreeBSD (<http://www.freebsd.org>), ein weiteres Unix-ähnliche Betriebssystem steht unter einer Lizenz, die genau dies ausdrücklich erlaubt – so enthalten etwa Windows NT und Mac OS-X wesentliche Teile des BSD-Codes. Obwohl seine Stabilität und Performance unstrittig sind, erreichte FreeBSD selbst nie die Popularität von GNU/Linux.

²² Vgl. *Rehbinder*, Urheberrecht, 9. Aufl., München (1996), S 62.

fundamentale Prinzip des Urheberrechts benutzt, um die sonst üblichen Beschränkungen geistigen Eigentums zu umgehen und dem Benutzer vier Freiheiten zu gewähren:

- Das Recht ein Programm zu benutzen.
- Das Recht auf freien Zugang zum Quellcode ohne zusätzliche Kosten.
- Das Recht, Kopien anzufertigen und – auch gegen Entgelt – weiterzugeben.
- Das Recht den Quellcode zu verändern, und auch diese veränderte Version weiterzuverbreiten, vorausgesetzt, dass auch die abgeleitete Version der GPL unterstellt wird.

Nach der Diktion der FSF ist eine Lizenz, welche die ersten drei Freiheiten einräumt, frei und das entsprechend lizenzierte Programm *freie Software*. Die vierte Freiheit, gleichzeitig die einzige, welche mit einer wirklichen Einschränkung verbunden ist, wird (nicht ohne Ironie) als *Copyleft* bezeichnet²³. Weiters enthalten die Ziffern 11. und 12. einen umfangreichen Haftungsausschluss, „soweit die geltenden Gesetze dies zulassen“. Die GPL verbietet es aber keineswegs, für die Überlassung von Programmen eine Gegenleistung zu verlangen. Was sie untersagt, ist die Einhebung eines Entgelts, das an Art, Intensität, Dauer oder an sonst einen auf die Benutzung bezogenen Faktor gekoppelt ist.²⁴ Dies mag auf den ersten Blick verwundern, ist aber genau die Existenzgrundlage von Unternehmen wie RedHat, SuSE und anderen kommerziellen Linux-Distributoren: Zwar besteht die Möglichkeit das gesamte System gratis aus dem Netz herunterzuladen, dies ist jedoch aufgrund der Downloadzeiten kostenintensiv und relativ umständlich: Die Installation verlangt einiges an Erfahrung. Distributoren bekommen ihr Entgelt also nicht als Lizenz- oder Benutzungsgebühr, sondern in erster Linie als Gegenleistung für die Zusammenstellung aller notwendigen Programme auf CDs und weiterer Serviceleistungen.

6. Ihre rechtliche Bewertung

Ein vorrangiges Ziel der GPL ist, den Quellcode eines Programms dauerhaft verfügbar zu halten. Dieses Ziel kann auf juristischem Weg nur erreicht werden, wenn ihr Rechtsverbindlichkeit zukommt und in der Fol-

²³ Software unter der GPL, wie der Linux-Kernel ist also Copyleft, FreeBSD dagegen „nur“ freie Software.

²⁴ Die GPL untersagt daher zB Softwaremiete.

ge bei Verstößen gerichtlich durchsetzbare Ansprüche bestehen, die hier im Detail nicht näher erörtert werden können.²⁵ Geprüft werden soll nur, an welchen gesetzlichen Vertragstypen die Überlassung von Software unter der GPL orientiert ist und welche Normen somit für die rechtliche Beurteilung überhaupt in Betracht kommen. Hier ist zunächst festzuhalten, dass die GPL für das angloamerikanische Copyright konzipiert wurde und offiziell lediglich in englischer Fassung vorliegt. Dennoch wird sie international eingesetzt, populärstes aber keineswegs einziges Beispiel ist der Linux-Kernel, der seinen Ursprung in Finnland hat²⁶. Das weltweit verbreitete Territorialprinzip im Bereich des Urheberrecht, zum Ausdruck gebracht auch im § 34 IPRG, besagt nun, dass Entstehen, Inhalt und Erlöschen von Immaterialgüterrechten nach dem Recht jenes Landes zu beurteilen sind, in dem eine Benutzungs- oder Verletzungshandlung gesetzt wird²⁷, weshalb der rechtliche Charakter einer Lizenzvereinbarung prinzipiell in jedem Land gesondert zu prüfen ist. Für die USA hat dies Lee²⁸ versucht, indem er die GPL mit zwei dort entschiedenen Fällen zur Gültigkeit von Shrink- bzw Click-Wrap Lizenzen²⁹ verglich. Nachdem das Bestehen einer Lizenzvereinbarung in beiden Fällen bejaht wurde, kommt er zu dem Schluss, dass die GPL, da sie im Gegensatz zu den entschiedenen Fällen den Nutzer ja nicht einschränke, sondern ihm im Gegenteil Freiheiten einräume, ebenfalls vor Gericht Bestand haben müsse. Kennt man jedoch die Praxis, so muss diese Folgerung voreilig erscheinen: Die GPL unterscheidet sich deutlich von Shrink- oder Click-Wrap Lizenzen. Denn niemand, der die Quellen des Linux-Kernels herunterlädt, kompiliert und installiert, ist dabei angehalten auf irgendeine Art zu bestätigen, dass er den Bedingungen der GPL zustimme; gleiches gilt für zahlreiche andere Programme unter der GPL. Für andere als englischsprachige Län-

²⁵ Dies ist Gegenstand der in Arbeit befindlichen Dissertation des Autors.

²⁶ Eine weitere Besonderheit von Linux ist auch , dass nicht (mehr) Linus Torvalds, wie er selbst betont, der alleinige Copyrightholder ist. Die Datei <http://www.kernel.org/pub/linux/kernel/CREDITS> ist eine Liste all jener Programmierer aus aller Welt, die wesentliche Beiträge geleistet haben, außerdem enthalten die einzelnen Kernel-Dateien, die nicht von Torvalds selbst erstellt wurden eine Angabe ihres Urhebers.

²⁷ Dazu im Detail Blocher, Gewerblicher Rechtsschutz und Urheberrecht, in: *Jahnel/Schramm/Staudegger* (Hrsg), Informatikrecht, Wien (2000), S 87.

²⁸ Lee, Open Source Software Licensing (Draft, 1999), im Netz unter <http://eon.law.harvard.edu/openlaw/gpl.pdf>.

²⁹ *ProCD, Inc vs Zeidenberg* (7th Cir. 1996), *Compuserve vs Patterson* (6th Cir. 1996). Beide Entscheidungen wurden in *Hotmail vs Van Money Pie Inc* bekräftigt (ND Cal, 1998).

der muss zudem die Frage gestattet sein, ob überhaupt von einer Zustimmung zu einer Lizenz ausgegangen werden kann, von der ausdrücklich nur eine Version als verbindlich gilt, nämlich die englische. Und im europäischen Raum wird die Gültigkeit von Shrink- und Click-Wrap Lizenzen generell bezweifelt³⁰. Trotzdem scheint die Geltung der GPL als urheberrechtliche Lizenz zB in Italien bis dato nicht in Zweifel gezogen zu werden.³¹ Dies ist auch insofern interessant, als gerade dort mit dem Gesetz 248/2000 eine Bestimmung in Geltung steht, die in doch sehr deutlichem Widerspruch zur Entwicklung von Software ganz allgemein (ob frei oder proprietär) steht, da sie eine Registrierungspflicht für jedermann, der zu Erwerbszwecken programmiert, vorschreibt.³² In Deutschland wird der Vertrieb von Software unter der GPL überwiegend als Schenkung gesehen. *Siepmann*³³ will die verminderte Haftung des Schenkers nach § 521 BGB sogar für kommerzielle Distributoren, die freie Software verkaufen zur Anwendung kommen lassen. *Jäger/Metzger*³⁴ nehmen zusätzlich eine Prüfung der GPL anhand des deutschen AGB-Gesetzes vor, welche ihrer Ansicht nach zum Wegfall des Haftungsausschlusses führt, schwächen dies jedoch wiederum ab, indem sie sich ebenfalls für eine Haftung nach Schenkungsrecht aussprechen. *Sester*³⁵ orientiert seine Be-

³⁰ Zur äußerst fraglichen Bindungswirkung solcher Lizenzen *Staudegger*, Rechtsfragen beim Erwerb von IT-Systemen in: *Jahnel/Schramm/Staudegger* (Hrsg), S 54; *Ertl*, Allgemeine Geschäftsbedingungen der Softwareverträge, EDVuR 1994, S 19. Speziell zu den sogenannten OEM-Lizenzen für vorinstallierte Software *Witte*, Urheberrechtliche Gestaltung des Vertriebs von Standardsoftware, CuR 1999, S 65; Vgl auch *Hoeren/Schumacher*, Verwendungsbeschränkungen im Softwarevertrag, CuR 2000, S 137. Auch nach britischem bzw schottischem Recht, das ja größere Ähnlichkeit zum amerikanischen Copyright als zum kontinentaleuropäischen Urheberrecht hat, wird die Verbindlichkeit solcher Lizenzen verneint, *Newton*, System Supply Contracts in: *Reed/Angel*, Computer Law, 4th Edition, London (2000), S 42.

³¹ *Di Salvatore*, I contratti informatici, Neapel (2000), S 88.

³² Zu Problematik dieser Bestimmung siehe <http://www.fsfeurope.org/siae/siae.it.html>.

³³ *Siepmann*, Lizenz- und haftungsrechtliche Fragen bei der kommerziellen Nutzung Freier Software JurPC Web-Dok. 163/1999, online unter <http://www.jura.uni-sb.de/jurpc/aufsatz/19990163.htm>.

³⁴ *Metzger/Jäger*, Open Source Software und deutsches Urheberrecht, GRUR Int 1999, S 839 sowie *Jäger*, GPL und Haftung: Ohne Verantwortung? Linux-Magazin 5/2000, S 134. Beide Aufsätze sind unter http://www.ifross.de/ifross_html/publikation.html zu finden.

³⁵ *Sester*, Open Source Software: Vertragsrecht, Haftungsrisiken und IPR-Fragen, CuR 2000, S 797.

urteilung am Recht der BGB-Gesellschaft³⁶, räumt aber ein, dass die sich daraus ergebenden Rechtsfolgen im wesentlichen unklar und erst zu erarbeiten seien.

Der OGH³⁷ hatte einen Vertrag über (entgeltliche) Softwareüberlassung auf die Anwendbarkeit von Gewährleistungsnormen zu prüfen. Dabei stellte er fest, dass urheberrechtliche Lizenzbestimmungen lediglich schuldrechtlich bindende Nebenbestimmungen zum jeweiligen Grundgeschäft, zB Miete oder Kauf seien. Eine Lizenz ist die Einräumung einer Nutzungserlaubnis an gewerblichen Schutzrechten, insbesondere an urheberrechtlich geschützten Werken, die als vertragliche Vereinbarung übereinstimmende Willenserklärungen voraussetzt³⁸. Doch um welche Art der Vereinbarung handelt es sich hier, wenn schon die Kenntniss der Lizenz nicht ohne weiteres vorausgesetzt werden kann? *Raymond* beschreibt die Gemeinschaft der Entwickler Freier Software als eine Geschenkkultur³⁹ – eine Gesellschaft, in der Reputation nicht mehr erworben wird durch Statussymbole in Form von Besitztümern, sondern durch den Wert dessen, was jemand gibt. Dieses Prinzip ist nicht so realitätsfern, wie es auf den ersten Blick scheinen mag, kann doch die Reputation, die durch die Teilnahme an einem freien Softwareprojekt erworben in der „echte Welt“ insoferne Umwegrentabilität entfalten, als sie dem Betroffenen zB zu einem lukrativeren Job verhilft. Trotz der Börsengänge von GNU/Linux-Unternehmen lässt sich also der Aspekt der Freigiebigkeit⁴⁰, nicht so einfach wegdiskutieren. Hier wird Software geschrieben und damit Wert geschaffen, dessen gegenwärtig erzielbarer Marktpreis nicht eingehoben wird. Die GPL, und sie ist schlussendlich die Willenserklärung dessen, der sie verwendet, stellt in ihrer Präambel ihr im Kern altruistische Ziel klar: „*To make sure the software is free for all of its users*“. Allerdings ist dies nicht ihr einziger Aspekt. Es darf wohl davon ausgegangen werden, dass sich das System GNU/Linux ohne die Beteiligung zahlreicher Programmierer aus aller Welt nicht so entwickelt hätte wie dies tatsächlich

³⁶ Zum sozietären Aspekt der GPL vgl schon *Jakob*, Being Free, zu finden unter http://www.users.sbg.ac.at/~jack/legal/being_free/being_free_html/being_free.html.

³⁷ 5Ob 504/96, 505/96 vom 14. 10. 1997=SZ 701202.

³⁸ *Schönherr*, Gewerblicher Rechtsschutz und Urheberrecht, Wien (1982), S 24 u 26; *Staudegger* in: *Jahnel/Schramm/Staudegger*, S 52; *Jakob*, Sponsoringverträge, Universität Salzburg (1999), S 74.

³⁹ *Raymond*, S 97.

⁴⁰ Und damit der Geschenkkarakter, vgl *Koziol/Welser*, Bürgerliches Recht, Bd. II, 11. Aufl., Wien (2000), S 168.

der Fall war. Aus verhandlungstheoretischer Sicht⁴¹ war es für *Torvalds*, als er am Beginn der Entwicklung von Linux stand wesentlich vernünftiger, seinen Quellcode frei im Internet zur Verfügung zu stellen, als zu versuchen, seinen noch rudimentären Kernel zu verkaufen. So konnte er *hoffen*⁴², für sein geben die Kooperation anderer Programmierer zu bekommen. Und *Stallman* schrieb kürzlich: „*Nobody is forced to join our club, but those who wish to participate must offer us the same cooperation they receive from us. That makes the system fair.*“⁴³ Es lassen sich also zwei Aspekte, jener der Freigiebigkeit und jener der Kooperation skizzieren, die in den folgenden Beziehungsfacetten in Erscheinung treten:

- Erstens zwischen den Entwicklern der Software (zB *Linus Torvalds*) und *allen* Benutzern (dh sowohl Distributoren als auch Endnutzern, die Programme herunterladen oder von Dritten erwerben).
- Zweitens zwischen den beteiligten Entwicklern untereinander. In diese Kategorie können auch Unternehmen fallen, die in freie Softwareprojekte investieren.
- Und drittens zwischen Distributoren (also Unternehmen die freie Software gegen Entgelt vertreiben) und ihren Kunden.

Im ersten Fall, also im Verhältnis zwischen den Entwicklern, die ihre Programme unter der GPL kostenlos anbieten und jenen, die davon profitieren, indem sie das Programm nur nutzen oder ohne eigene Änderungen weiterverbreiten, dominiert das Element der Freigiebigkeit, also die schenkungsrechtliche Komponente. Zwar erfolgt ein Lizenzieren unter der GPL häufig in der Erwartung, weitere Programmierer zur Mitarbeit zu gewinnen, doch wird auf dieser Stufe nicht versucht, dieser Hoffnung im Vertragswege auch *rechtliche* Durchschlagskraft zu verleihen. Der *erklärte* Wille, nicht nur sich selbst, sondern eben auch die Gegenseite zu binden, ist jedoch die Grundvoraussetzung jedes nicht unentgeltlichen Geschäfts. Bereits hier von einer gesellschaftsrechtlichen Vertragsbeziehung auszugehen scheint aber auch deshalb verfehlt, weil wie oben gezeigt, in der Praxis weder die Kenntnis der Lizenz noch ein ausdrückliche Zustimmung zu ihr nötig ist, um ein Programm nutzen zu können. Zwar ist auch die Schenkung ein zustimmungsbedürftiges Rechtsgeschäft, doch

⁴¹ Vgl *Graf*, Vertrag und Vernunft, Wien (1999), S 68.

⁴² Wohlgemerkt: Eine synallagmatische Beziehung im Sinne des *do ut des* besteht hier (noch) nicht.

⁴³ *Stallman*, The GNU GPL and the American Way, zu finden unter: [http://weblog.mercurycenter.com/ejournal/stories/storyReader\\$664](http://weblog.mercurycenter.com/ejournal/stories/storyReader$664).

erlaubt die Tatsache, dass der Lizenznehmer im Falle der GPL ausschließlich begünstigt wird, solange er den Quellcode nicht verändert *und* diese Änderungen weiterverbreitet, die Wirksamkeitsanforderungen an eine konkludente Zustimmung⁴⁴ nach unten zu setzen und eine solche schon im Benutzen der Software zu sehen. Selbst nach deutschem Recht verzichtbar erscheint iS dieses teleologischen Verständnisses aber eine Inhaltskontrolle der GPL als AGB. Denn eine solche Inhaltskontrolle fußt auf Rechtsschutzüberlegungen⁴⁵ des Gesetzgebers, die wiederum dort nicht gegeben sind, wo jener Teil, der geschützt werden soll nur begünstigt wird. Und der Lizenznehmer wird, solange er nur nützt, privat verändert oder unverändert weiterverbreitet durch die GPL keinen Einschränkungen unterworfen. Anders stellt sich die Lage nur dar, wenn der Lizenznehmer das Programm verändert *und* diese Veränderung oder das modifizierte Gesamtprogramm weiterverbreitet. Um ein Programm ändern zu können, muss der Programmierer die Quelldateien, also Textdateien öffnen. Spätestens hier finden sich jedoch schon in den ersten Kommentarzeilen Hinweise auf die GPL, so dass nun sehr wohl von einer Kenntnis der Lizenzierung ausgegangen werden kann. Das Recht, ein Werk in verändertem Zustand weiterverbreiten zu dürfen, ist das weitestgehende Recht, welches einem Werknutzer überhaupt eingeräumt werden kann. Jemand, der vorhat, ein Programm so weitgehend zu nutzen, darf also nicht ohne weiteres von der Zulässigkeit dieses Handelns ausgehen. Von ihm darf auch erwartet werden, dass er sich näher über die Einzelheiten der Lizenzierung informiert und ggf eine Übersetzung besorgt. Dies legt es nahe, spätestens in der ersten Weiterverbreitungshandlung von Modifikationen die Zustimmung zur Verpflichtung zu sehen, auch den Quellcode weiterzugeben. Da über diesen zweifelsfrei klar wird, dass ein Programm unter der GPL steht und auch die Einholung näherer Informationen zumutbar ist, wäre ein nachträglicher Verstoß jedenfalls ein *venire contra factum proprium*. Im Zusammenhang mit dem Offenlegen von Virusdefinitionen hat *Mayer-Schönberger*⁴⁶ gezeigt, dass hier der Beitrag zu einer gemeinsamen Zielerreichung⁴⁷, einem gemeinsamen Nutzen im Vordergrund steht.

⁴⁴ Zu den ansonsten strengen Anforderungen *Koziol/Welser*, Bd. I, 11. Aufl., Wien (2000), S 91.

⁴⁵ Vgl zum österreichischen Konsumentenschutzrecht *Koziol/Welser*, Bd.II, S 363.

⁴⁶ *Mayer-Schönberger*, Taming Lakatos' Monster, Vortrag am 4.11. 2000 auf der WebNet, San Antonio (bisher unveröffentlicht).

⁴⁷ *Jabornegg* in: *Schwimann* (Hrsg), Praxiskommentar zum ABGB, Bd IV/2, § 1175, Rz 12; *Kastner*, Grundriß des Österreichischen Gesellschaftsrechts, 5. Aufl., Wien (1990), S 68; *Schmidt*, Gesellschaftsrecht, 3. Aufl., Köln (1995), S 64.

Der Softwareentwickler wird also durch das Veröffentlichen seiner Modifikationen Partner in einer kooperativen Vertragsbeziehung, die ihn zur Öffnung seines Quellcodes verpflichtet und in der nun über den gemeinsame Zweck, ein Programm zu schaffen und zu verbessern Elemente der Gesellschaft Bürgerlichen Rechts in den Vordergrund treten. Dies gilt umso mehr dann, wenn sich Unternehmen durch die Investition von Know-How oder finanziellen Mitteln an freien Softwareprojekten beteiligen. Ein Beispiel für eine sich unter Umständen daraus ergebende, konkrete Rechtsfolge ist das Verbot einer Kündigung zur Unzeit in § 1212 ABGB⁴⁸, welches in einer kritischen Pre-Release Phase einem Programmierer oder Investor den Ausstieg aus einem Softwareprojekt verwehren könnte. Erwirbt aber ein (End-) Benutzer eine GNU/Linux-Distribution, wie sie in vielen Elektrogrossmärkten und Buchhandlungen angeboten wird, so schließt er mit dem Veräußerer einen Kaufvertrag, auf den prinzipiell die Bestimmungen der §§ 1053 ff ABGB und die Gewährleistungsregeln⁴⁹ anzuwenden sind ab. Dieser Kaufvertrag besteht zwischen dem Käufer und Verkäufer, das wird häufig nicht der eigentliche Ersteller der Distribution sein. Bei letzterem wäre eine Haftung nach dem PHG denkbar⁵⁰, die ggf den Importeur treffen würde. Hier stellt sich die Frage, ob der Haftungsausschluss der Ziffern 11. und 12. GPL auch insoweit Geltung beanspruchen kann, als er *alle* Weiterverbreiter einschließt.

Die Lizenzierung eines Programms unter der GPL kann daher als Angebot mit zwei rechtlichen Dimensionen interpretiert werden: Einerseits ein schenkungsähnliches Verhältnis, das durch die bloße Benutzung (konkludent) angenommen wird, andererseits eine kooperative Beziehung, die durch die erste Verbreitung veränderter Programmversionen eingegangen wird. Im zweiten Fall wäre es allerdings verfehlt, von einem bloß ideellem Gesellschaftszweck auszugehen: Wie gesagt, liegt der gemeinsame Zweck in der Erstellung qualitativ hochwertiger Software, deren Quellcode frei ist bzw bleibt. Dass dabei kein Endprodukt verkauft wird, bedeutet noch nicht, dass die Entwickler keinen finanziellen Vorteil aus dem Projekt lukrieren. Denn schon der Einsatz leistungsfähigerer Software im eigenen Unternehmen kann erhebliche finanzielle Vorteile bringen und im eBusiness bedeutet jede Minute, die ein Server verfügbar

⁴⁸ Vgl *Kastner*, S 73.

⁴⁹ SZ 70/202.

⁵⁰ Zur Haftung für Software nach dem PHG *Musulas*, Die Haftung für Software im Hinblick auf das ProdHaftG, Frankfurt (1993), S 21.

ist, bares Geld. So wird die Entwicklung des Webservers Apache⁵¹ bis heute in erster Linie von Webmastern getragen. Als IBM sich dann am Apache-Projekt durch die Beisteuerung von Quellcode beteiligte, geschah auch dies aus handfesten wirtschaftlichen Überlegungen: Denn damit konnte die unrentable Produktion des eigenen Webservers eingestellt und gleichzeitig ein neuer Markt erschlossen werden⁵². Es besteht hier also sehr wohl das Ziel der Erwirtschaftung und Verteilung eines Vertragsgewinns⁵³, nur eben in einer bisher unüblichen Weise, wobei freie Softwarelizenzen⁵⁴ für die hier verfolgten Ziele offenbar das effizienteste Regelwerk statuieren.

7. Where do we want to go tomorrow?

Höchst interessant ist die Tatsache, dass trotz der pragmatischen Ausrichtung der Open Source Initiative die GPL in der letzten Zeit auch bei den Größten der EDV-Branche immer mehr Anklang findet. Sun Microsystems veröffentlichte sein StarOffice Paket als OpenOffice⁵⁵ unter der GPL, das Datenbanksystem MySQL wurde unter der GPL freigegeben und IBM veröffentlicht neben seinem beachtlichen finanziellen Engagement in unterschiedliche freie Softwareprojekte seit kurzem auch Hardwaretreiber unter dieser Lizenz⁵⁶. Der wichtigste Grund dafür liegt wohl darin, dass die GPL, obwohl – oder gerade weil – sie noch niemals einer richterlichen Prüfung unterzogen war, eine auch wirtschaftlich relevante Sicherheit gibt, wie kaum eine andere Lizenz. Denn wenn ein Unternehmen schon bereit ist, Know-How offenzulegen, so wird es vermeiden wollen, dass ein Konkurrent davon auf dem Markt ohne Gegenleistung profitiert. Während zB die FreeBSD License die Repropriarisierung fördert, minimiert die GPL dieses Risiko, indem sie hier sozusagen zur Zusammenarbeit zwingt und den Wettbewerb auf die Service- und Imageebene verlagert. Der Trend in diese Richtung wird durch die negativen Erfahrungen mit der Propriarisierung und Aufspaltung von Unix verstärkt, der den einzelnen Unternehmen kleine Marktsegmente mit relativ

⁵¹ <http://www.apache.org>.

⁵² *Moody*, S 289.

⁵³ Vgl *Graf*, S 83.

⁵⁴ Apache steht nicht unter der GPL, sondern unter einer der FreeBSD License ähnelnden Lizenz.

⁵⁵ <http://www.openoffice.org>.

⁵⁶ http://oss.software.ibm.com/developer/opensource/linux/projects/_mwave/.

niedrigen Gewinnaussichten aber gleichzeitig hohen Entwicklungskosten bescherte, so dass man sich hier nach offenen und einheitlichen Standards geradezu sehnt.

Es mag so scheinen, als sei GNU/Linux ein Beispiel dafür, wie eine effektive Selbstregulierung im Internet tatsächlich funktionieren kann. Ob dies dauerhaft der Fall sein wird, kann nur die Zeit zeigen. Bemerkenswert ist jedenfalls, wie es hier gelingt, einen freien Diskurs nicht nur auf kognitiver sondern auch auf produktiver Ebene zu führen und wie der Erfolg der Bazar-Methode bereits auf andere Bereiche als den der Software abzufärben beginnt: So hat etwa das Opencontent-Projekt⁵⁷ zwei Lizenzen erarbeitet, welche die Verbreitung von Literatur unter ähnlichen Bedingungen wie freie Software ermöglichen sollen und das Berkman Center for Internet and Society in Harvard⁵⁸ versucht mit Openlaw ein ähnliches Modell für die gemeinsame Bearbeitung von Rechtsfällen zu etablieren. Eine solche Modellfunktion wird aber nur dann tatsächlich übernommen werden können, wenn man einerseits begreift, dass freie Software nichts mit der Abschaffung geltenden Urheberrechts zu tun hat, sondern gerade auf dessen Prinzipien aufbaut. Zum anderen ist zu hoffen, dass zumindest die europäischen Rechtssetzungsorgane dem Druck einiger Softwarekonzerne und der Unterhaltungsindustrie kritischer gegenüberstehen als jene in den USA und es zumindest vermeiden, dieser grundsätzlich positiv zu beurteilenden Entwicklung unnötige Steine⁵⁹ in den Weg zu legen.

⁵⁷ <http://www.opencontent.org>.

⁵⁸ <http://www.openlaw.org>.

⁵⁹ Ein Beispiel dafür sind etwa die Restriktionen bezüglich des Fair Use, die der Digital Millennium Copyright Act enthält und seine zT heftig kritisierte Interpretation durch das erstinstanzliche Gericht im DVD-Linking Fall (zum aktuellen Stand siehe <http://eon.law.harvard.edu/openlaw/DVD/>).