

Martin Schulze  
Palmstr. 4  
30419 Hannover

Hannover, 10.07.2003

# **Das Spannungsverhältnis von Open Source Software und Softwarepatenten nach deutschem Recht**

Betreuer: Prof. Dr. Wolfgang Kilian

---

Abschlussarbeit im Rahmen des Ergänzungsstudiengangs  
Rechtsinformatik am Fachbereich Rechtswissenschaften der  
Universität Hannover

EULISP VII

Juli 2003

# Inhaltsverzeichnis

<b>I. Einleitung</b> .....	<b>1</b>
<b>II. Software als Gegenstand eines urheberrechtlichen und patentrechtlichen Schutzes</b> .....	<b>2</b>
1. Schutz von Software nach dem Urheberrecht.....	2
2. Schutz von Software nach dem Patentrecht.....	4
3. Das Verhältnis von urheberrechtlichen und patentrechtlichen Schutz.....	5
<b>III. Die Patentierbarkeit von Software „de lege lata“</b> .....	<b>5</b>
1. Voraussetzungen des § 1 PatG.....	6
a. Technische Erfindung.....	6
aa. Grundlegendes zum Begriff der Technik.....	7
bb. Die Technizität von Software und das Patentierungsverbot für Programme für Datenverarbeitungsanlagen „als solche“, § 1 II Nr. 3, III PatG.....	7
(1) Stand der Rechtsprechung.....	8
(a) Rechtsprechung des Bundesgerichtshofs.....	9
(aa) Kerntheorie.....	9
(bb) Gesamtbetrachtungslehre.....	14
(b) Neuere Rechtsprechung des Bundespatentgerichts.....	18
(c) Ergebnis.....	20
(2) Meinungsstand in der Literatur.....	23
(3) Eigene Stellungnahme.....	24
(4) Vereinbarkeit mit Art. 27 TRIPS Übereinkommen.....	30
b. Neuheit, § 1 i.V.m. § 3 PatG.....	32
c. Erfinderische Tätigkeit, § 1 i.V.m. § 4 PatG.....	33
d. Gewerbliche Anwendbarkeit, § 1 i.V.m. § 5 PatG.....	34
e. Sonderfall: Computerbezogene Geschäftsmethoden.....	35
2. Aktuelle Entwicklungen auf Europäischer Ebene.....	36
a. Revision des Europäischen Patentübereinkommens (EPÜ 2000).....	36
b. Vorschlag für eine EU-Richtlinie zur Patentierbarkeit von computer-implementierten Erfindungen.....	37
aa. Vorbereitende Schritte.....	37
bb. Aufbau und Regelungsinhalt.....	38
(1) Technizität einer computer-implementierten Erfindung.....	39
(2) Voraussetzungen der Patentierbarkeit.....	39
(3) Form des Patentanspruchs.....	40

(4) Regelungen zur Dekompilierung und Interoperabilität .....	41
cc. Ergebnis.....	41
<b>IV. Das Open Source Software Modell.....</b>	<b>42</b>
1. Die Entstehungsgeschichte der Open Source Bewegung.....	42
a. Der Bedeutungswandel von Software zu einem eigenständigen Wirtschafts-	
gut und der Niedergang der Hackerkultur.....	42
b. Das UNIX Betriebssystem.....	44
c. Das GNU/Linux-Projekt.....	46
d. Die wirtschaftliche Bedeutung von Open Source Software .....	47
2. Zum Begriff der Open Source Software.....	49
a. Die Open Source Software Definition.....	50
b. Open Source Software vs. Freie Software .....	51
c. Open Source Lizenzmodelle.....	53
aa. Copyleft Lizenzen.....	53
bb. Non-Copyleft Lizenzen.....	54
cc. Dual Licensing.....	54
d. Abgrenzung zu anderen Vertriebsformen.....	55
aa. Proprietäre Software.....	55
bb. Freeware.....	55
cc. Shareware.....	56
dd. Public Domain Software.....	56
3. Rechtliche Ausgestaltung einer Open Source Lizenz am Beispiel der GNU	
General Public License.....	57
<b>V. Das Spannungsverhältnis von Open Source Software und Software-</b>	
<b>patenten.....</b>	<b>58</b>
1. Einführung in die allgemeine rechtspolitische Diskussion um Softwarepatente .....	59
2. Spezifische Gefahren für Entwicklung, Vertrieb und Nutzung von Open Source	
Software durch Softwarepatente.....	60
a. Verletzung von Patenten proprietärer Software.....	60
aa. Verletzung von Erzeugnis- und Verfahrenspatenten.....	62
bb. Ausnahmeregelung des § 11 Nr. 1 und Nr. 2 PatG.....	63
cc. Rechtsfolgen.....	64
b. Unberechtigte Patentanmeldungen für proprietäre Software.....	65
aa. Veröffentlichung von Eigenentwicklungen.....	65
bb. Einspruch und Nichtigkeitsklage.....	67
cc. Geltendmachung eines Vorbenutzungsrechts gemäß § 12 I PatG.....	68

c. Anmeldung von Patenten auf Open Source Software zwecks proprietärer Vermarktung.....	68
d. Ergebnis.....	71
3. Lösungsvorschläge zur Verringerung des strukturellen Ungleichgewichts zwischen dem Open Source Modell und der proprietären Softwareentwicklung...	72
a. Softwarepatenten im Allgemeinen.....	73
aa. Verbesserung der Prüfungsverfahren und Recherchemöglichkeiten.....	73
bb. Verkürzung der Laufzeit von Softwarepatenten.....	74
b. Open Source Software spezifische Lösungsvorschläge.....	74
aa. Erweiterung der Ausnahmetatbestände des § 11 PatG.....	75
(1) Privilegierung der nichtgewerblichen Nutzung außerhalb des privaten Bereichs.....	75
(2) Quelltextprivileg.....	76
bb. Erteilung von Zwangslizenzen gemäß § 24 I PatG .....	76
cc. Open Patents Pool und Einführung einer Neuheitsschonfrist.....	77
<b>VI. Fazit und Ausblick.....</b>	<b>80</b>

## Literaturverzeichnis

- Anders, Wilfried Die Patentierbarkeit von Programmen für Datenverarbeitungsanlagen: Rechtsprechung im Fluß?  
GRUR 1989, S. 861 – 868
- Anders, Wilfried Wie viel technischen Charakter braucht eine computer-implementierte Geschäftsmethode, um auf erfinderischer Tätigkeit zu beruhen?  
GRUR 2001, S. 555 - 560
- Anders, Wilfried Patentierbare Computerprogramme  
GRUR 1990, S. 498 – 500
- Benkhard, Georg Patentgesetz  
Gebrauchsmustergesetz  
9. Auflage  
München 1993  
(zit.: Benkhard-Bearbeiter, PatG)
- Betten, Jürgen Anmerkung zu BGH, Urt. v. 11.03.1986  
GRUR 1986, S. 534 – 535
- Betten, Jürgen Anmerkung zu BGH, Beschl. v. 11.05.2000  
GRUR 2000, S. 1009 – 1010
- Betten, Jürgen Patentschutz von Computerprogrammen  
GRUR 1995, S. 775 – 789
- Busche, Jan Softwarebezogene Erfindungen in der Entscheidungspraxis des Bundespatentgerichts und des Bundesgerichtshofs  
Mitt. 2001, S. 49 – 57
- Busche, Jan Der Schutz von Computerprogrammen – Eine Ordnungsaufgabe für Urheberrecht und Patentrecht?  
Mitt. 2000, S. 164 – 173
- Busse, Rudolf Kommentar zum Patentgesetz  
5. Auflage  
Berlin 1999  
(zit.: Busse-Bearbeiter, PatG)

- Deike, Theis  
Open Source Software: IPR-Fragen und Einordnung ins deutsche Rechtssystem  
CR 2003, S. 9 – 18
- Engel, Friedrich-Wilhelm  
Über „Computerprogramme als solche“  
GRUR 1993, S. 194 – 199
- Esslinger, Alexander/  
Betten, Jürgen  
Patentschutz im Internet  
CR 2000, S. 18 – 22
- Esslinger, Alexander/  
Hössle, Markus  
Zur Entscheidung „State Street v. Signature Financial“ des amerikanischen Court of Appeals für the Federal Circuit  
Mitt. 1999, S. 327 – 329
- Gehring, Robert  
Software Patents – IT-Security at Stake?  
22. Februar 2002  
<http://ig.cs.tu-berlin.de/ap/rg/2001-10/Gehring2001Full-SWPatITSec.pdf><sup>1</sup>  
(zit.: Gehring, IT-Security)
- Gehring, Robert  
Freeware, Shareware und Public Domain  
Berlin 1996  
<http://ig.cs.tu-berlin.de/sa/043/>  
(zit.: Gehring, Freeware, Shareware und PD)
- Goebel, Frank Peter  
Technizität – zum Patentschutz für programmbezogene Erfindungen nach der jüngeren deutschen Erteilungs- und Entscheidungspraxis  
S. 357 – 378  
in: Festschrift für Rudolf Nirk zum 70. Geburtstag  
Herausgegeben von Karl Bruchhausen, Wolfgang Hefermehl, Peter Hommelhoff, Herbert Messer  
München 1992  
(zit.: Goebel, FS-Nirk)
- Grassmuck, Volker  
Freie Software  
Zwischen Privat- und Gemeineigentum  
Bonn 2002  
<http://freie-software.bpb.de/Grassmuck.pdf>  
(zit.: Grassmuck, Freie Software)
- Grzeszick, Bernd  
Freie Software: Eine Widerlegung der Urheberrechtstheorie?  
MMR 2000, S. 412 – 417

---

<sup>1</sup> Hinweis: Sämtliche in dieser Arbeit aufgeführten Links befinden sich auf dem Stand vom 08.07.2003.

Hausberg, Bernhard	Zur Einführung der Neuheitsschonfrist im Patentrecht – ein USA-Deutschland-Vergleich bezogen auf den Hochschulbereich  Studie im Auftrag des Bundesministeriums für Bildung und Forschung (BMBF) November 2001 (zit.: Hausberg, BMBF-Studie zur Neuheitsschonfrist)
Heide, Nils	Softwarepatente im Verletzungsprozess CR 2003, S. 165 – 171
Horns, Axel H.	Der Patentschutz für softwarebezogene Erfindungen im Verhältnis zur „Open Source“-Software JurPC Web-Dok. 223/2000, Abs. 1 – 80
Horns, Axel H.	Anmerkungen zu begrifflichen Fragen des Softwareschutzes GRUR 2001, S. 1 – 21
Hössle, Markus	Der nicht-statische Technikbegriff Mitt. 2000, S. 343 – 346
Howard, Anthony	Patentability of Computer-Implemented Inventions CRI 2002, S. 97 – 104
Hufnagel, Frank-Erich	Software – und Business-Patente – Herausforderung für das juristische Risikomanagement MMR 2002, S. 279 – 283
Jaeger, Till/ Metzger, Axel	Open Source Software Rechtliche Rahmenbedingungen der Freien Software München 2002 (zit.: Jaeger/Metzger, OSS)
Jänich, Volker Michael	Sonderrechtsschutz für geschäftliche Methoden GRUR 2003, S. 483 – 489
Kindermann, Manfred	Softwarepatentierung (I) Stand der Rechtsprechung des BGH und EPA CR 1992, S. 577 – 588

- Kindermann, Manfred  
 Softwarepatentierung (II)  
 Stand der Rechtsprechung des BPatG,  
 Rechtsprechungsvergleich und Schluß-  
 folgerungen  
 CR 1992, S. 658 – 666
- Klopmeier, Felix  
 Zur Technizität von Software  
 Mitt. 2002, S. 65 – 70
- Koch, Frank A.  
 Urheber- und kartellrechtliche Aspekte der  
 Nutzung von Open-Source-Software (I)  
 CR 2000, S. 273 – 281
- Koch, Frank A.  
 Urheber- und kartellrechtliche Aspekte der  
 Nutzung von Open-Source-Software (II)  
 CR 2000, S. 333 – 344
- Kolle, Gert  
 Technik, Datenverarbeitung und Patentrecht  
 GRUR 1977, S. 58 – 74
- König, Reimar  
 Patentfähige Datenverarbeitungsprogramme –  
 ein Widerspruch in sich  
 GRUR 2001, S. 577 – 584
- Lehmann, Michael  
 Rechtsschutz und Verwertung von  
 Computerprogrammen  
 2. Auflage  
 Köln 1993  
 (zit.: Bearbeiter in: Lehmann, Rechtsschutz  
 von Computerprogrammen)
- Lutterbeck, Bernd/  
 Gehring, Robert/  
 Horns, Axel H.  
 Sicherheit in der Informationstechnologie und  
 Patentschutz für Software-Produkte – ein  
 Widerspruch?  
 Kurzgutachten im Auftrag des Bundes-  
 ministeriums für Wirtschaft und Technologie  
 Dezember 2000  
 (zit.: Lutterbeck/Gehring/Horns,  
 Kurzgutachten)
- Marly, Jochen  
 Urheberrechtsschutz für Computersoftware in  
 der Europäischen Union  
 München 1995  
 (zit.: Marly, Computersoftware in der EU)

- Melullis, Klaus-J. Zum Patentschutz für Computerprogramme  
In Festschrift für Willi Erdmann zum 65.  
Geburtstag  
Herausgegeben von Hans-Jürgen Ahrens,  
Joachim Bornkamm, Wolfgang Gloy, Joachim  
Starck und Joachim v. Ungern-Sternberg  
Köln 2002  
S. 401 – 423  
(zit.: Melullis, FS-Erdmann)
- Melullis, Klaus-J. Zur Patentfähigkeit von Programmen für  
Datenverarbeitungsanlagen  
GRUR 1998, S. 843 – 853
- Mes, Peter Patentgesetz  
Gebrauchsmustergesetz  
München 1997  
(zit.: Mes, PatG)
- Metzger, Axel Softwarepatente im künftigen europäischen  
Patentrecht  
CR 2003, S. 313 – 317
- Metzger, Axel/  
Jaeger, Till Open Source Software und deutsches  
Urheberrecht  
GRUR Int. 1999, S. 839 – 848
- Milbradt, Claudia Schutzfähigkeit von Software  
K&R 2002, S. 522 – 526
- Moglen Eben Free Software or Open Source?  
Dezember 2000  
[http://moglen.law.columbia.edu/publications/  
lu-07.pdf](http://moglen.law.columbia.edu/publications/lu-07.pdf)
- Morgan, Jason V. Chaining Open Source Software: The Case  
Against Software Patents  
1999  
<http://lpf.ai.mit.edu/Patents/chaining-oss.html>  
(zit.: Morgan, Chaining OSS)
- Nack, Ralph Sind jetzt computerimplementierte  
Geschäftsmethoden patentfähig?  
GRUR Int. 2000, S. 853 – 858

- Nack, Ralph/  
Straus, Joseph
- Mikro- und makroökonomische Implikationen der Patentierbarkeit von Softwareinnovationen: Geistige Eigentumsrechte in der Informationstechnologie im Spannungsfeld von Wettbewerb und Innovation
- Forschungsprojekt des Max-Planck-Instituts und dem Fraunhofer ISI im Auftrag des Bundesministeriums für Wirtschaft und Technologie  
September 2001  
(zit.: Nack/Straus, MPI-Gutachten)
- Ochmann, Richard
- Zum Begriff der Erfindung als Patentschutzvoraussetzung  
S. 759 - 775  
in: Festschrift für Rudolf Nirk zum 70. Geburtstag  
Herausgegeben von Karl Bruchhausen, Wolfgang Hefermehl, Peter Hommelhoff, Herbert Messer  
München 1992  
(zit.: Ochmann, FS-Nirk)
- Ohly, Ansgar
- Software und Geschäftsmethoden im Patentrecht  
CR 2001, S. 809 – 817
- Omsels, Hermann-Josef
- Open Source und das deutsche Vertrags- und Urheberrecht  
S. 141 – 170  
in: Festschrift für Paul W. Hertin zum 60. Geburtstag  
Herausgegeben von Christian Schertz und Hermann-Josef Omsels  
München 2000  
(zit.: Omsels, FS-Hertin)
- Palandt, Otto
- Bürgerliches Gesetzbuch  
62. Auflage  
München 2003  
(zit.: Palandt-Bearbeiter)
- Pierson, Matthias
- Der Schutz der Programme für die Datenverarbeitung im System des Immaterialgüterrechts  
Pfaffenweiler 1991  
(zit.: Pierson, Der Schutz der Programme für die Datenverarbeitung)

Raubenheimer, Andreas	Die jüngere BGH-Rechtsprechung zum Softwareschutz nach Patentrecht CR 1994, S. 328 – 335
Röttinger, Moritz	Patentierbarkeit computerimplementierter Erfindungen CR 2002, S. 616 – 619
Sandl, Ulrich	„Open Source“-Software: Politische, ökonomische und rechtliche Aspekte CR 2001, S. 346 – 351
Schar, Markus	Zum objektiven Technikbegriff im Lichte des Europäischen Patentübereinkommens Mitt. 1998, S. 322 – 339
Schiffner, Thomas	Open Source Software Freie Software im deutschen Urheber- und Vertragsrecht München 2002 (zit.: Schiffner, OSS)
Schiama, Daniele	TRIPS und das Patentierungsverbot von Software „als solcher“ GRUR Int. 1998, S. 852 – 858
Schmidtchen, Jürgen	Zur Patentfähigkeit und zur Patentwürdigkeit von Computerprogrammen und von programmbezogenen Lehren Mitt. 1999, S. 281 – 294
Schölch, Günther	Softwarepatente ohne Grenzen GRUR 2001, S. 16 – 21
Sedlmaier, Roman	Verwirrung durch Klarstellungen im Softwarepatentrecht Mitt. 2002, S. 55 – 65
Sedlmaier, Roman	Der Richtlinienvorschlag für die Patentierbarkeit computerimplementierter Erfindungen – eine Anmerkung Mitt. 2002, S. 97 – 101
Sester, Peter	Open-Source-Software: Vertragsrecht, Haftungsrisiken und IPR-Fragen CR 2000, S. 797 – 807

Siepmann, Jürgen	Lizenz- und haftungsrechtliche Fragen bei der kommerziellen Nutzung Freier Software JurPC Web-Dok. 163/1999, Abs. 1 – 289
Spindler, Gerald	Rechtsfragen der Open Source Software Studie im Auftrag des Verbandes der deutschen Softwareindustrie e.V. (VSI) 2003 <a href="http://www.vsi.de/inhalte/aktuell/studie_final.pdf">http://www.vsi.de/inhalte/aktuell/studie_final.pdf</a> (zit.: Spindler, Rechtsfragen der OSS)
Straus, Joseph	Bedeutung des TRIPS für das Patentrecht GRUR 1996, S. 179 – 205
Tauchert, Wolfgang	Patentschutz für Computerprogramme – Sachstand und neue Entwicklungen GRUR 1999, S. 829 – 833
Tauchert, Wolfgang	Zum Begriff der „technischen Erfindung“ JurPC Web-Dok. 28/2002, Abs. 1 – 59
Tauchert, Wolfgang	Zur Patentierbarkeit von Programmen für Datenverarbeitungsanlagen Mitt. 1999, S. 248 – 252
Tauchert, Wolfgang	Zum Begriff der „technischen Erfindung“ JurPC Web-Dok. 28/2002, Abs. 1 – 59
Ullmann, Eike	Urheberrechtlicher und patentrechtlicher Schutz von Computerprogrammen CR 1992, S. 641 – 648
van Raden, Lutz	Die Informativische Taube GRUR 1995, S. 451 – 458
von Hellfeld, Axel	Sind Algorithmen schutzfähig? GRUR 1989, S. 471 – 485
Winischhofer, Thomas	Computersoftware und Patentrecht Diss. 2000 <a href="http://www.winischhofer.net/other/patent.pdf">http://www.winischhofer.net/other/patent.pdf</a> (zit.: Winischhofer, Computersoftware und Patentrecht)
Wuermerling, Ulrich/ Deike, Theis	Open Source Software: Eine juristische Risikoanalyse CR 2003, S. 87 – 91



## **I. Einleitung**

*„Versuche, Patentschutz für Software-Schöpfungen zu erhalten, haben keine Zukunft mehr.“, Gert Kolle 1977<sup>1</sup>*

*„Von jetzt an [bestehen] keine substanziellen Schranken für die Patentfähigkeit von Softwareerfindungen.“, Ralph Nack, 2000<sup>2</sup>*

Zwischen diesen beiden Einschätzungen liegen knapp 25 Jahre und dennoch Welten. Beide Zitate entstammen Anmerkungen zu zwei Urteilen des BGH zur Patentierbarkeit von Software. Sie verdeutlichen, dass in der nunmehr fast 30-jährigen Diskussion um die Patentierbarkeit von Software ein erheblicher Entwicklungsprozess stattgefunden hat. Allerdings könnte sich auch der falsche Eindruck aufdrängen, dass sich die Diskussion um die Patentierbarkeit von Software nunmehr geklärt hätte. Tatsächlich aber ist um die Frage der Patentierbarkeit von Software nach wie vor eine erhebliche Kontroverse im Gange. Während lange Zeit der urheberrechtliche Schutz von Computersoftware im Vordergrund stand, hat insbesondere der Siegeszug des Internets und die damit einhergehende Eröffnung von völlig neuen Geschäftsfeldern, die Frage nach einem patentrechtlichen Schutz von Software wieder ins Rampenlicht der rechtlichen und öffentlichen Diskussion geführt. Während sich in den USA relativ früh eine liberale Haltung gegenüber Softwarepatenten entwickelte, schritt die Entwicklung in Europa und insbesondere in Deutschland eher zögerlich voran. Erst in jüngster Zeit, ist es, nicht zuletzt aufgrund intensiver Harmonisierungsbemühungen seitens der EU, auch in Europa wieder zu einer heftigen Auseinandersetzung mit diesem Thema gekommen. Hier ist es insbesondere die Open Source Bewegung, welche Softwarepatente als eine unmittelbare Bedrohung für die Zukunft ihres Entwicklungsmodells betrachtet. Zuweilen drängt sich hier der Eindruck auf, es ginge dabei um den Kampf um Sein oder Nichtsein. In der Tat werden die besonderen Gefahren, welche mit der Patentierung von Software verbunden sind nirgendwo anders so deutlich wie in dem Verhältnis zu Open Source Software. Während das Open Source Software Modell auf einem freien und uneingeschränkten Wissenstransfer aufbaut, fördern Softwarepatente

---

<sup>1</sup> Kolle, GRUR 1977, 58 (74).

<sup>2</sup> Nack, GRUR Int. 2000, 853 (858).

die Exklusivität geistigen Eigentums wie kein anderes Schutzinstrument. Gegenstand dieser Arbeit soll das besondere Spannungsverhältnis zwischen Open Source Software und Softwarepatenten sein. Dabei soll untersucht werden, ob und inwiefern Softwarepatente eine Gefahr für die Entwicklung, den Vertrieb und die Nutzung von Open Source Produkten darstellen, und welche Möglichkeiten bestehen, eventuell vorhandene Risiken zu vermeiden oder wenigstens einzudämmen. Interessant ist dabei vor allem die Frage, ob es Mittel und Wege gibt, diese gegenläufigen und scheinbar völlig unvereinbaren Konzepte zu einer ausgeglichenen Koexistenz zu bewegen. Bevor auf diese Frage einzugehen ist, soll jedoch zunächst der unterschiedliche Schutz von Software nach dem Urheber- und dem Patentrecht dargestellt werden (Abschnitt II). In einem zweiten Schritt ist die gegenwärtige Rechtslage zur Patentierbarkeit von Software in Deutschland und insbesondere die dazu vorliegende Rechtsprechung des BGH darzustellen (Abschnitt III). In Abschnitt IV sollen die entwicklungstechnischen und lizenzrechtlichen Besonderheiten des Open Source Software Modells dargestellt werden, um dann abschließend die Auswirkungen von Softwarepatenten auf das Open Source Software Modell zu untersuchen (Abschnitt V).

## **II. Software als Gegenstand eines urheberrechtlichen und patentrechtlichen Schutzes**

Die Frage der Patentierbarkeit von Software kann nicht isoliert betrachtet werden. Um die höchst kontroverse Diskussion über die Patentierbarkeit von Computerprogrammen angemessen zu beleuchten, ist es erforderlich den unterschiedlichen Schutz von Software nach dem Urheberrecht und dem Patentrecht aufzuzeigen.

### **1. Der Schutz von Software nach dem Urheberrecht**

Die Diskussion um den urheberrechtlichen Schutz von Computerprogrammen reicht zurück bis in die 1960er Jahre.<sup>3</sup> Nach einigen, sich teilweise widersprechenden instanzgerichtlichen Entscheidungen wurde die urheberrechtliche Schutzfähigkeit von Computerprogrammen vom BGH in seiner „*Inkasso-Programm*“

---

<sup>3</sup> Vgl. dazu *Marly*, Computersoftware in der EU, S. 73ff.; *Pierson*, Der Schutz der Programme für die Datenverarbeitung, S. 46ff; *Schiffner*, OSS, S. 40ff.

Entscheidung vom 09.05.1985 grundsätzlich anerkannt.<sup>4</sup> Nur wenige Monate später wurde der urheberrechtliche Schutz von Computerprogrammen als Sprachwerke in § 2 I Nr. 1 UrhG a.F. auch gesetzlich verankert.<sup>5</sup> Nach den vom BGH in der „*Inkasso-Programm*“ Entscheidung aufgestellten und später in der „*Betriebssystem*“ Entscheidung<sup>6</sup> vom 04.10.1990 bestätigten Grundsätzen, wurden jedoch hohe Anforderungen an die erforderliche Schöpfungshöhe von Computerprogrammen gestellt. Nur solche Programme, die ein bestimmtes Maß an schöpferischer Individualität erkennen ließen und eine über das allgemeine Durchschnittskönnen in bedeutendem Maße hinausgehende Gestaltungstätigkeit in Auswahl, Sammlung, Anordnung und Einteilung der Informationen und Anweisungen aufwiesen, sollten urheberrechtlichen Schutz genießen.<sup>7</sup> Diese extrem hohen Anforderungen führten im Ergebnis dazu, dass nur einem verschwindend geringen Anteil von Programmen tatsächlich ein urheberrechtlicher Schutz zuerkannt wurde. Zu einer erheblichen Ausweitung des urheberrechtlichen Schutzes von Computerprogrammen kam es jedoch durch die Umsetzung der EG-Software-Richtlinie<sup>8</sup> durch Gesetz vom 09.06.1993.<sup>9</sup> Der Schutz von Computerprogrammen wurde durch die §§ 69a ff. UrhG erheblich erweitert. Gemäß § 69a III UrhG kommt es nunmehr für den Schutz eines Computerprogramms lediglich darauf an, dass dieses Ergebnis einer eigenen geistigen Schöpfung des Urhebers darstellt. Weitere qualitative oder ästhetische, insbesondere eine besondere Gestaltungshöhe sind nicht mehr erforderlich. Im Ergebnis genießt daher jedes Computerprogramm urheberrechtlichen Schutz, sofern es ein Mindestmaß an Individualität (sog. kleine Münze) aufweist.<sup>10</sup> Das Urheberrecht schützt dabei die konkrete Ausdrucksform eines Computerprogramms so wie es sich in seiner linguistischen Gestaltung als persönliche geistige Schöpfung darstellt. Erfasst wird nur die äußere Ausdrucksform des Programms, nicht jedoch die zugrunde liegende Idee oder die Funktionalität desselben. Der Schutz

---

<sup>4</sup> BGH GRUR 1985, 1041ff. – „*Inkasso-Programm*“.

<sup>5</sup> Gesetz zur Änderung von Vorschriften auf dem Gebiet des Urheberrechts vom 24. Juni 1985, BGBl. I 1985, S. 1137.

<sup>6</sup> BGH GRUR 1991, 449ff. – „*Betriebssystem*“.

<sup>7</sup> BGH GRUR 1985, 1041 (1047); BGH GRUR 1991, 449 (451).

<sup>8</sup> Richtlinie 91/250/EWG des Rates vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen, Abl. EG Nr. L 122, S. 42.

<sup>9</sup> Zweites Gesetz zur Änderung des Urheberrechtsgesetzes vom 09. Juni 1993, BGBl. I 1993, S. 910

<sup>10</sup> *Jaeger/Metzger*, OSS, S. 20; *Schiffner*, OSS, S. 46; *Ullmann*, CR 1992, 641 (643).

erstreckt sich gemäß § 69c UrhG im Wesentlichen auf das unberechtigte Verbreiten, Vervielfältigen und Bearbeiten eines Computerprogramms. Es ist urheberrechtlich jedoch ohne weiteres zulässig, die Funktionalität eines bestimmtes Computerprogramms mit eigenen Mitteln nachzuprogrammieren.

## **2. Der Schutz von Software nach dem Patentrecht**

Ein Patent ist ein zeitlich und territorial beschränktes Monopol, welches dem Erfinder gegen Offenbarung seiner Erfindung vom Staat verliehen wird, sofern diese neu und gewerblich anwendbar ist, und auf einer erfinderischen Tätigkeit beruht.<sup>11</sup> Das Patent soll dem Erfinder die ausschließliche gewerbliche Verwertung seiner Erfindung für bestimmte Zeit sichern. Die Schutzdauer beträgt gemäß § 16 I PatG zwanzig Jahre, beginnend mit dem Tag, der auf die Anmeldung der Erfindung folgt. Im Gegensatz zum Urheberrecht schützt das Patentrecht nicht das konkrete linguistische Konzept eines Computerprogramms, sondern die durch den Programmcode ausgedrückte Funktionalität desselben. Geschützt wird also die dem Programm zugrunde liegende Idee. Damit reicht der durch ein Patent gewährte Schutz wesentlich weiter als der durch das Urheberrecht vermittelte Schutzbereich. Gemäß § 9 S. 1 PatG, hat der Patentinhaber das ausschließliche Recht die patentierte Erfindung zu benutzen, d.h. ein Erzeugnis, das Gegenstand des Patents ist, herzustellen, anzubieten, in Verkehr zu bringen oder zu gebrauchen (§ 9 S. 2 Nr. 1 PatG), ein Verfahren, das Gegenstand des Patents ist, anzuwenden oder anzubieten (§ 9 S. 2 Nr. 2 PatG) oder das durch ein Verfahren, das Gegenstand des Patents ist, unmittelbar hergestellte Erzeugnis anzubieten, in Verkehr zu bringen oder zu Gebrauchen (§ 9 S. 2 Nr. 3 PatG). Bezogen auf ein Computerprogramm bedeutet dies, dass der Schutzbereich des Patents nicht etwa dadurch umgangen werden kann, dass ein Programm mit identischer Funktion neu programmiert wird. Sämtliche Programme, welche dieselbe Funktionalität wie das patentierte Computerprogramm aufweisen, stellen ein Verletzung des Patents dar. Gemäß § 11 Nr. 1 und Nr. 2 PatG erstreckt sich die Wirkung des Patents jedoch nicht auf rein private Handlungen und Handlungen, welche zu Versuchszwecken

---

<sup>11</sup> Vgl. zu den Voraussetzungen im Einzelnen unten III. 1.

vorgenommen werden.<sup>12</sup> Verletzt jemand die Rechte aus den §§ 9ff. PatG, so kann der Patentinhaber gemäß § 139 PatG Unterlassungs- und Schadensersatzansprüche geltend machen.

### **3. Verhältnis von urheberrechtlichen und patentrechtlichen Schutz**

Wie sich aus den obigen Ausführungen ergibt, erstreckt sich der urheberrechtliche und der patentrechtliche Schutz auf zwei völlig unterschiedliche Aspekte eines Computerprogramms. Während das Urheberrecht die konkrete Ausdrucksform eines Computerprogramms unabhängig von der dadurch realisierten Funktionalität schützt, betrifft das Patentrecht die Funktionalität eines Programms, unabhängig von der gewählten Ausdrucksform. Daraus ergibt sich, dass urheberrechtlicher und patenrechtlicher Schutz nebeneinander stehen können.<sup>13</sup> Dies entspricht dem im gewerblichen Rechtsschutz und im Immaterialgüterrecht vorherrschenden Kumulationsprinzip<sup>14</sup> und wird bestätigt durch § 69g I UrhG, wonach die Anwendung sonstiger Rechtsvorschriften, unter anderem über den Schutz für Erfindungen, durch die besonderen urheberrechtlichen Vorschriften der §§ 69a ff. UrhG unberührt bleibt.

### **III. Die Patentierbarkeit von Software „de lege lata“**

Im Folgenden soll untersucht werden, wie sich die Patentierbarkeit von Software nach geltendem Recht darstellt. Hierbei soll zum einen eine kritische Auseinandersetzung mit der zu dieser Frage vorliegenden Rechtsprechung und Literatur erfolgen. Außerdem sind auch die jüngsten Entwicklungen auf europäischer Ebene zu untersuchen. Der Begriff „Softwarepatent“ mag den Eindruck erwecken, dass es dabei nur um die Patentierung eines Computerprogramms an sich geht. Die Patentierung eines reinen „Computerprogrammprodukts“ stellt aber nur einen, bisher in der Erteilungspraxis der Patentämter und der Rechtsprechung eher seltenen Fall eines Softwarepatents dar. Unter dem Begriff „Softwarepatent“ werden vielmehr sämtliche, als Verfahrens- oder

---

<sup>12</sup> Inwiefern diese Ausnahmetatbestände für die Entwicklung und Nutzung von Open Source Software nutzbar gemacht werden können, wird unter V.2.a.bb. untersucht.

<sup>13</sup> *Busche*, Mitt. 2000, 164 (165f.); *Goebel*, FS-Nirk, S. 377; *Horns*, GRUR 2001, 1 (5); *Milbradt*, K&R 2002, 522 (526); *Raubenheimer*, CR 1994, 328 (334); *Ullmann*, CR 1992, 641 (647f.).

<sup>14</sup> *Schiurma*, GRUR Int. 1998, 852 (855).

Erzeugnispatente (vgl. § 9 PatG) angemeldete Erfindungen zusammengefasst, die zu ihrer Realisierung einen in bestimmter Weise programmierten Universalrechner erfordern.<sup>15</sup> Dazu zählen im Übrigen auch die nicht minder umstrittenen Patente auf elektronische Geschäftsmethoden, wie z.B. die in den USA erteilten Patente für Amazon.com's 1-Click-Verfahren<sup>16</sup> oder Priceline.com's Reverse Auction System<sup>17</sup>. Im Folgenden soll daher der inhaltlich umfassendere Begriff der computerbezogenen Erfindungen verwendet werden.

### 1. Voraussetzungen des § 1 PatG

Gemäß § 1 I PatG<sup>18</sup> werden Patente für Erfindungen erteilt, die neu sind, auf einer erfinderischen Tätigkeit beruhen und gewerblich anwendbar sind. Während die Frage, ob überhaupt eine Erfindung vorliegt, als ein Kriterium der Patentfähigkeit einzuordnen ist, stellen die Voraussetzungen der Neuheit, der erfinderischen Tätigkeit und der gewerblichen Anwendbarkeit Kriterien der Patentwürdigkeit dar.

#### a. Technische Erfindung

Patente werden nur für Erfindungen erteilt. Eine Legaldefinition einer Erfindung findet sich im PatG nicht, stattdessen wird in § 1 II PatG eine Negativliste von Gegenständen und Tätigkeiten aufgeführt, welche nicht als Erfindungen im Sinne von Abs. 1 anzusehen sind.<sup>19</sup> Völlig unstrittig ist jedoch, dass, obwohl das Gesetz dies nicht ausdrücklich anordnet, nur solche Erfindungen patentfähig sind, die dem Gebiet der Technik angehören.<sup>20</sup> Dies ergibt sich zum einen aus dem Zusammenspiel von § 1 I PatG mit den §§ 3 und 4 PatG, in denen jeweils auf den Stand der Technik abgestellt wird. Zum anderen gilt das Technizitätserfordernis aber auch als ein dem Patentrecht seit jeher innewohnender und damit gewohnheitsrechtlich gefestigter Grundsatz.<sup>21</sup> Nach Ansicht des BGH begründet das

---

<sup>15</sup> Vgl. *Horns*, JurPC Web-Dok. 223/2000, Abs. 5.

<sup>16</sup> US Patent Nr. 5.960.411; vgl. dazu *Jänich*, GRUR 2003, 483 (484).

<sup>17</sup> US Patent Nr. 5.897.620.

<sup>18</sup> In der Fassung der Bekanntmachung vom 16.12.1980, BGBl. I 1981, S. 1.

<sup>19</sup> Die Negativliste wurde eingefügt durch das Gesetz über das Internationale Patentübereinkommen vom 21.06.1976, BGBl. II 1976, S. 648.

<sup>20</sup> BGH GRUR 1977, 152 (153) – „Kennungsscheibe“; BGH GRUR 1977, 96 (98) – „Dispositionsprogramm“; *Benkhard-Bruchhausen*, PatG, § 1, Rn. 45; *Busse-Keukenschrijver*, PatG, § 1, Rn. 14; *Schulte*, PatG, § 1, Rn. 22; *Tauchert*, JurPC Web-Dok. 28/2002, Abs. 1; *van Raden*, GRUR 1995, 451 (454).

<sup>21</sup> *Kolle*, GRUR 1977, 58 (61f.); *Ochmann*, FS-Nirk, S. 768.

Erfordernis des technischen Charakters das einzig brauchbare Abgrenzungskriterium gegenüber anderen Leistungen des Menschen, für die ein Patentschutz weder vorgesehen noch geeignet ist.<sup>22</sup>

#### **aa. Grundlegendes zum Begriff der Technik**

Der Begriff der Technik wird im PatG allerdings nicht näher definiert. Eine grundlegende Definition des Begriffs der Technik wurde vom BGH erstmals in seiner „Rote Taube“ Entscheidung aufgestellt.<sup>23</sup> Technisch ist danach eine „Lehre zum planmäßigen Handeln unter Einsatz beherrschbarer Naturkräfte zur Erreichung eines kausal übersehbaren Erfolgs“.<sup>24</sup> Diese Definition wurde vom BGH in einer Reihe von weiteren Entscheidungen bestätigt und teilweise präzisiert. Zum einen stellte der BGH klar, dass die menschliche Verstandeskraft selbst nicht zu den beherrschbaren Naturkräften zählt.<sup>25</sup> Zum anderen stellte er klar, dass der kausal übersehbare Erfolg die *unmittelbare* Folge des Einsatzes beherrschbarer Naturkräfte sein muss.<sup>26</sup> Zu einem späteren Zeitpunkt hat der BGH seinen Technikbegriff insoweit fortentwickelt, als auf den Einsatz beherrschbarer Naturkräfte dann verzichtet werden kann, wenn die beanspruchte Lehre eine Lösung ersetzt, die auf Mitteln der traditionellen Technik beruht.<sup>27</sup> Insgesamt ist der Technikbegriff des Patentrechts also nicht statisch, sondern Modifikationen zugänglich, sofern die technologische Entwicklung und ein daran angepasster effektiver Patentschutz dies erfordern.<sup>28</sup>

#### **bb. Die Technizität von Software und das Patentierungsverbot für Programme für Datenverarbeitungsanlagen „als solche“, § 1 II Nr. 3, III PatG**

Nach der aktuellen Rechtslage sind Programme für Datenverarbeitungsanlagen „als solche“ gemäß § 1 II Nr. 3, III PatG vom Patentschutz ausgenommen. Diese Regelung wurde wortwörtlich aus Art. 52 II c) und III EPÜ übernommen. Über die Bedeutung dieser Regelung besteht schon seit jeher erhebliche Kontroverse, sowohl in

<sup>22</sup> BGH GRUR 1977, 96 (99) – „Dispositionsprogramm“; BGH GRUR 1992, 36 (38) – „Chinesische Schriftzeichen“.

<sup>23</sup> BGH GRUR 1969, 672 – „Rote Taube“.

<sup>24</sup> BGH GRUR 1969, 672 (673).

<sup>25</sup> BGH GRUR 1977, 96 (98) – „Dispositionsprogramm“; BGH GRUR 1977, 152 (153) – „Kennungsscheibe“; BGH GRUR 1980, 849 (850) – „Antiblockiersystem“.

<sup>26</sup> BGH GRUR 1977, 152 (153) – „Kennungsscheibe“.

<sup>27</sup> BGH GRUR 2000, 498 (501) – „Logikverifikation“.

<sup>28</sup> So ausdrücklich BGH GRUR 2000, 498 (501); vgl. auch *Hössle*, Mitt. 2000, 343ff.

der Rechtsprechung als auch in der juristischen Literatur. Während sich in der Literatur um unterschiedlichste Interpretationen bemüht wurde, vermied die Rechtsprechung es lange Zeit, zu der gesetzlichen Regelung ausdrücklich Stellung zu nehmen. Stattdessen stützte sich die Rechtsprechung zur Abgrenzung von patentierfähigen und nicht patentierfähigen Computerprogrammen auf das allgemeine Kriterium der Technizität. Erst in seiner jüngsten Entscheidung vom 17.10.2001 („*Suche fehlerhafter Zeichenketten*“) äußerte sich der BGH zum Patentierungsverbot des § 1 II Nr. 3, III PatG.<sup>29</sup> Der Senat stellt zunächst fest, dass sich aus dem Zusammenspiel von § 1 II Nr. 3 und III PatG ergibt, dass weder Programme für Datenverarbeitungsanlagen schlechthin vom Patentschutz ausgenommen sind, noch das bei Vorliegen der weiteren Voraussetzungen des Gesetzes für jedes Computerprogramm Patentschutz erlangt werden kann.<sup>30</sup> Zur Abgrenzung zwischen nicht patentierfähigen Computerprogrammen „als solche“ und patentierfähigen computerbezogenen Erfindungen stellt der BGH sodann darauf ab, ob die beanspruchte Lehre der Lösung eines konkreten technischen Problems dient.<sup>31</sup> Mit dieser Interpretation des Patentierungsverbots für Computerprogramme setzt der BGH seine bisherige Rechtsprechung zu computerbezogenen Erfindungen fort, in denen er seit jeher auf das Kriterium der Technizität zurückgegriffen hat, um zwischen patentierfähigen und nicht patentierfähigen computerbezogenen Erfindungen zu unterscheiden.

### **(1) Stand der Rechtsprechung**

Im Folgenden soll anhand der bisherigen Rechtsprechung des BGH und den neueren Entscheidungen des BPatG zu computerbezogenen Erfindungen das Merkmal der Technizität von Computerprogrammen, welches der BGH nunmehr auch ausdrücklich zur Konkretisierung des Patentierungsverbots des § 1 II Nr. 3, III PatG heranzieht, umfassend dargestellt und beleuchtet werden.

---

<sup>29</sup> BGH CR 2002, 88ff. – „*Suche fehlerhafter Zeichenketten*“. In seinem Urteil vom 13.12.1999 - „*Logikverifikation*“ hatte der BGH diese Frage noch ausdrücklich offen gelassen (BGH GRUR 2000, 498 (501)).

<sup>30</sup> BGH CR 2002, 88 (90).

<sup>31</sup> BGH CR 2002, 88 (90).

### **(a) Rechtsprechung des Bundesgerichtshofs**

Die Rechtsprechung des BGH ist vor allem durch zwei unterschiedliche Betrachtungsweisen gekennzeichnet. In den 70er und 80er Jahren dominierte die Betrachtung nach der sog. Kerntheorie. Einer computerbezogenen Erfindung kam danach technischer Charakter zu, wenn der als neu und erfinderisch anzusehende Kern der beanspruchten Lehre auf technischem Gebiet liegt.<sup>32</sup> Anfang der 90er Jahre rückte der BGH von der Kerntheorie ab und stützte sich, in Anlehnung an die Rechtsprechung der Beschwerdekammern des EPA, fortan auf eine Gesamtbetrachtung der beanspruchten Lehre (sog. Gesamtbetrachtungslehre). Danach kommt es für die Beurteilung der Technizität einer computerbezogenen Erfindung auf eine wertende Betrachtung dahingehend an, ob technische Elemente bei der beanspruchten Lehre im Vordergrund stehen.<sup>33</sup>

#### **(aa) Kerntheorie**

In seiner ersten Entscheidung zur Frage der Patentierbarkeit eines Computerprogramms aus dem Jahre 1976 („*Dispositionsprogramm*“) musste sich der BGH mit einem Anspruch auf ein Verfahren zur Ermittlung von Veränderungen einer Vielzahl beliebiger Größen mit Hilfe einer zum Stand der Technik gehörenden elektronischen Datenverarbeitungsanlage mit bestimmten Hardwarekomponenten befassen.<sup>34</sup> Das BPatG hatte das Patent mit der Begründung abgelehnt, der Verfahrensgegenstand sei ein reiner Rechen- und Organisationsplan, dem als bloßer Anweisung an den menschlichen Geist kein technischer Charakter zukomme. Auch der BGH verneinte den technischen Charakter des Verfahrens und nutzte die Gelegenheit um grundlegende Ausführungen zur Frage der Technizität eines Computerprogramms zu machen. Dabei stützte sich der BGH zunächst auf die grundlegende Technikdefinition aus der „*Rote Taube*“ Entscheidung, wonach als patentierbar eine Lehre zum planmäßigen Handeln unter Einsatz beherrschbarer Naturkräfte zur Erreichung eines kausal übersehbaren Erfolgs anzusehen ist.<sup>35</sup> Zwar sei auch die in Rede stehende Organisations- und Rechenregel als eine Anweisung zum planmäßigen Handeln zur Erreichung eines

---

<sup>32</sup> Vgl. z.B. BGH GRUR 1978, 102 – „Prüfverfahren“.

<sup>33</sup> Vgl. z.B. BGH GRUR 2000, 498 (500) – „Logikverifikation“.

<sup>34</sup> BGH GRUR 1977, 96ff. – „Dispositionsprogramm“.

<sup>35</sup> BGH GRUR 1969, 672 – „Rote Taube“.

kausal übersehbaren Erfolgs anzusehen, es fehle hingegen am Einsatz beherrschbarer Naturkräfte, noch setze diese zur Bewältigung der gestellten Aufgabe deren Einsatz notwendig voraus.<sup>36</sup> Zur Bejahung der Patentierbarkeit eines Computerprogramms müsse der Einsatz technischer Mittel jedoch notwendiger Bestandteil der Problemlösung selbst sein.<sup>37</sup> Der BGH führt aus, dass ein Computerprogramm nur dann patentfähig sei, wenn das Programm einen neuen, erfinderischen Aufbau einer solchen Anlage erfordere oder wenn in ihm die Anweisung enthalten sei, die Anlage auf eine neue, bisher nicht übliche und auch nicht nahe liegende Art und Weise zu nutzen.<sup>38</sup> Problematisch ist hier jedoch, wie zwischen dem bestimmungsgemäßen Gebrauch einer Datenverarbeitungsanlage und einer neuen Brauchbarkeit zu differenzieren ist.<sup>39</sup> Zu bedenken ist auch, dass ein Universalrechner prinzipiell jede Aufgabe erfüllen kann, so dass es zweifelhaft erscheint, ob überhaupt eine neuer erfinderischer Aufbau gefunden werden kann.<sup>40</sup>

Bei der „Straken“ Entscheidung aus dem Jahre 1977 ging es um ein computergestütztes Verfahren zur Berechnung der Oberfläche von dreidimensionalen Körpern.<sup>41</sup> In Fortsetzung der „Dispositionsprogramm“ Entscheidung sah der BGH in dem Verfahren lediglich den bestimmungsgemäßen Gebrauch einer hinreichend bekannten Datenverarbeitungsanlage und verneint demzufolge den technischen Charakter des Programms. Der BGH verdeutlicht zudem, dass eine im Prinzip untechnische Lehre, wie die dem Straken zugrunde liegende Rechenregel, nicht etwa dadurch technisch werde, dass der *Nutzeffekt der Erfindung* auf einem technischen Gebiet liege.<sup>42</sup> In der noch im selben Jahr ergangenen Entscheidung „Prüfverfahren“ ging es um ein Verfahren zur Prüfung des Programmablaufs einer Datenverarbeitungsanlage.<sup>43</sup> Der BGH sah sich zwar mangels hinreichender tatsächlicher Feststellungen zu einer abschließenden Bewertung nicht im Stande, machte jedoch erstmals ausdrücklich klar, dass es bei der Beurteilung des technischen Charakters darauf

---

<sup>36</sup> BGH GRUR 1977, 96 (98).

<sup>37</sup> BGH GRUR 1977, 96 (98).

<sup>38</sup> BGH GRUR 1977, 96 (97).

<sup>39</sup> So auch *Kindermann*, CR 1992, 577 (581).

<sup>40</sup> *Kolle*, GRUR 1977, 58 (73).

<sup>41</sup> BGH GRUR 1977, 657ff. – „Straken“.

<sup>42</sup> BGH GRUR 1977, 657 (658).

<sup>43</sup> BGH GRUR 1978, 102ff. – „Prüfverfahren“.

ankomme, auf welchem Gebiet *der erfinderische Kern* der beanspruchten Lehre liege.<sup>44</sup> In der Entscheidung „Fehlerortung“ aus dem Jahre 1978 ging es um ein Verfahren zur Ortung von Fehlern in einer Datenverarbeitungsanlage.<sup>45</sup> Das BPatG hatte von den fünf beanspruchten Merkmalen der angemeldeten Erfindung nur in einem den eigentlichen Kern des Erfindungsgedankens gesehen. Diesen wiederum wertete das Gericht als untechnisch und verneinte demzufolge die Patentierbarkeit der Lehre insgesamt. Der BGH bestätigte in seiner Entscheidung die vom BPatG vorgenommene Betrachtung nach dem Kern der Lehre. Der BGH bekräftigte nochmals sein bereits in den vorhergegangenen Entscheidungen gefundenes Ergebnis, wonach eine generell untechnische Rechen- oder Organisationsregel nicht dadurch technisch werde, dass ihre Anwendung zweckmäßig oder auch ausschließlich unter Zuhilfenahme technischer Geräte erfolgt.<sup>46</sup>

Mit der Entscheidung „Antiblockiersystem“ aus dem Jahre 1980 bejahte der BGH erstmals den technischen Charakter eines auf ein Computerprogramm gestützten Verfahrens.<sup>47</sup> Inhaltlich ging es dabei um ein aus mechanischen und elektronischen Elementen bestehendes Verfahren zur Regelung druckmittelbetätigter Fahrzeugbremsen. Der BGH bejahte, entgegen der Ansicht des BPatG, den technischen Charakter des Verfahrens, weil Zweck des Antiblockiersystems die unmittelbare Erreichung einer optimalen Bremswirkung durch den planmäßigen Einsatz berechen- und beherrschbarer Naturkräfte ohne Zwischenschaltung menschlicher Verstandeskraft sei.<sup>48</sup> Der BGH unterschied hier erstmals zwischen Programmen technischer Natur und Programmen untechnischer Natur.<sup>49</sup> Insbesondere Programme aus dem Bereich der Regel- und Messtechnik, welche Messergebnisse aufarbeiten, den Ablauf technischer Einrichtungen überwachen oder sonst steuernd bzw. regelnd nach außen wirken, könnten als technische Programme ohne weiteres patentiert werden.<sup>50</sup>

---

<sup>44</sup> BGH GRUR 1978, 102.

<sup>45</sup> BGH GRUR 1978, 420ff. – „Fehlerortung“.

<sup>46</sup> BGH GRUR 1978, 420 (421).

<sup>47</sup> BGH GRUR 1980, 849ff. – „Antiblockiersystem“.

<sup>48</sup> BGH GRUR 1980, 849 (850).

<sup>49</sup> BGH GRUR 1980, 849 (851).

<sup>50</sup> BGH GRUR 1980, 849 (851).

Dieses Ergebnis wurde aber bereits in der darauf folgenden Entscheidung „Walzstabteilung“ vom 16.12.1980 wieder in Frage gestellt.<sup>51</sup> Es ging um ein Verfahren zum automatisierten Unterteilen von Walzstäben auf einer kontinuierlich arbeitenden Walzstrasse, wobei die Teillängen von einem mit Messeinrichtungen ausgestatteten Rechner derart berechnet werden sollten, dass jeweils ganzzahlige Vielfache einer Verkaufslänge entstehen sollten. Der BGH lehnte die Patentierbarkeit des Verfahrens ab, weil der erfinderische Kern, nämlich das dem Verfahren zugrunde liegende Programm, als reines Denkschema keinen technischen Charakter habe.<sup>52</sup> Daran änderte nach Ansicht des BGH auch die Tatsache nichts, dass das Programm aufgrund der gewonnenen Ergebnisse unmittelbar steuernd in den Ablauf eines Produktionsprozesses eingriff, da insoweit keine neue und erfinderische Verwendung der Produktionsanlage vorliege.<sup>53</sup> In Anbetracht der zuvor in der Entscheidung „Antiblockiersystem“ dargelegten Grundsätze zu technischen Programmen, ist die vom BGH getroffene Entscheidung nicht recht nachzuvollziehen.<sup>54</sup> Ganz im Zeichen der Kerntheorie stand auch die Entscheidung „Flugkostenminimierung“ vom 11.03.1986.<sup>55</sup> Gegenstand der Anmeldung war ein automatisches Verfahren zur optimalen Regelung des Treibstoffdurchsatzes eines Flugzeugs. Das Verfahren setzte sowohl beherrschbare Naturkräfte, als auch markt- und betriebswirtschaftliche Faktoren ein, um den erstrebten Erfolg einer Flugkostenminimierung zu realisieren. Nach Ansicht des BGH standen die markt- und betriebswirtschaftlichen Faktoren dabei im Vordergrund der beanspruchten Lehre und bildeten damit den Kern der Erfindung.<sup>56</sup> Auch dieses Urteil wirkt, angesichts des offensichtlich technischen Charakters des gesamten Verfahrens, befremdlich und offenbart deutlich die Schwächen der Kerntheorie. Bedenkt man, dass das gesamte Verfahren ohne den Einsatz des Rechners überhaupt keinen Sinn machen würde, erscheint es schwer nachvollziehbar, warum ausgerechnet die markt- und betriebswirtschaftlichen Faktoren den erfinderischen Kern der Lehre bilden sollen.<sup>57</sup>

---

<sup>51</sup> BGH GRUR 1981, 39ff. – „Walzstabteilung“.

<sup>52</sup> BGH GRUR 1981, 39 (41).

<sup>53</sup> BGH GRUR 1981, 39 (41).

<sup>54</sup> Kritisch auch *Busche*, Mitt. 2001, 49 (52); *Kindermann*, CR 1992, 577 (583).

<sup>55</sup> BGH GRUR 1986, 531ff. – „Flugkostenminimierung“.

<sup>56</sup> BGH GRUR 1986, 531 (533f.).

<sup>57</sup> So auch *Betten*, GRUR 1986, 534; *Kindermann*, CR 1992, 577 (584).

Mit der Entscheidung „*Seitenpuffer*“ aus dem Jahre 1991 leitete der BGH erstmals eine Korrektur seiner mittlerweile 15 Jahre alten Rechtsprechung zu Softwarepatenten ein.<sup>58</sup> Die nach der angemeldeten Lehre zu lösende Aufgabe bestand darin, den sog. Cachespeicher des Rechners so effizient zu nutzen, dass möglichst wenig Zugriffe auf den langsameren Hauptspeicher des Rechners notwendig waren. Das BPatG hatte in Anwendung der Grundsätze der „*Dispositionsprogramm*“ Entscheidung die Patentierbarkeit des Programms verneint, weil es sich um untechnisches Programm handele, welches lediglich in bestimmungsgemäßer Art und Weise von einer Datenverarbeitungsanlage Gebrauch mache. Der BGH hingegen bejahte die Technizität des Programms. Dabei erweiterte er in Ergänzung zu seiner Rechtsprechung aus der „*Dispositionsprogramm*“ Entscheidung den Kreis technischer Computerprogramme. Auch solche Programme, welche die Funktionsfähigkeit einer Datenverarbeitungsanlage als solche betreffen und damit das unmittelbare Zusammenwirken ihrer Elemente ermöglichen, seien technischer Natur.<sup>59</sup>

In der Entscheidung „*Chinesische Schriftzeichen*“ vom 11.06.1991 musste sich der BGH erstmals mit der Neufassung des PatG von 1981 beschäftigen.<sup>60</sup> Die Lehre betraf ein Verfahren zur Eingabe chinesischer Schriftzeichen in ein Textverarbeitungssystem mit einer Eingabetastatur, einer Steuer- und Speichereinheit und einer Anzeigeeinheit. Ziel des Verfahrens war vor allem eine schnellere und unter reduziertem Speicherplatzbedarf arbeitende automatische Erfassung des Bedeutungsgehalts der chinesischen Schrift, bei der eine hohe Anzahl an Zeichen eine Vielzahl von Bedeutungen haben kann. Der BGH verneinte den technischen Charakter des Verfahrens unter strikter Anwendung der Kerntheorie damit, dass der erfinderische Kern der Lehre lediglich ein gedanklich-logisches Ordnungssystem darstelle.<sup>61</sup> Insgesamt kann die Entscheidung des BGH nicht überzeugen, da die Bedeutung der technischen Komponenten des Verfahrens vollkommen vernachlässigt wurde.<sup>62</sup>

---

<sup>58</sup> BGH GRUR 1992, 33ff. – „*Seitenpuffer*“.

<sup>59</sup> BGH GRUR 1992, 33 (35).

<sup>60</sup> BGH GRUR 1992, 36ff. – „*Chinesische Schriftzeichen*“.

<sup>61</sup> BGH GRUR 1992, 36 (38).

<sup>62</sup> Kritisch auch *Betten*, GRUR 1995, 775 (786); *Busche*, Mitt. 2001, 49 (53f.); *Goebel*, FS-Nirk, S. 378; *Kindermann*, CR 1992, 577 (580); *Schmidtchen*, Mitt. 1999, 281 (285); *Raubenheimer*, CR 1994, 328 (331).

Ohne den Einsatz des Rechners wäre auch das gedanklich-logische Ordnungssystem, insbesondere unter dem Gesichtspunkt der gesteigerten Arbeitsgeschwindigkeit, nicht zu realisieren gewesen.

### **(bb) Gesamtbetrachtungslehre**

In der Entscheidung „*Tauchcomputer*“ aus dem Jahre 1992 rückt der BGH endgültig von seiner Kerntheorie ab.<sup>63</sup> Gegenstand der Anmeldung war eine Anzeigevorrichtung für die Parameter eines Tauchgangs, wie z.B. die aktuelle Tauchtiefe und die bisherige Tauchzeit. Zusätzlich verfügte die Anzeigevorrichtung über einen Speicher für die Dekompressionsparameter diverser Tauchtiefen und -zeiten, welche mit den gemessenen Werten des Tiefen- und Zeitmessers derart verknüpft war, dass jeweils die optimale Dekompression für den aktuellen Tauchgang angezeigt wurde. Dazu griff der Tauchcomputer auch auf bekannte Tauchtabellen zurück. Das BPatG hatte den technischen Charakter des Verfahrens mit der Begründung verneint, dass Kern der Lehre lediglich ein untechnisches Denkschema in Form einer besonderen Interpretation und Auswertung bekannter Tauchtabellen sei. Der BGH hingegen bejahte den technischen Charakter des Verfahrens. Dabei betonte er nunmehr, dass bei der Prüfung von Erfindungen, die Merkmale technischer Natur mit Merkmalen nichttechnischer Art verknüpfen, der gesamte Erfindungsgegenstand unter Einschluss der etwaigen Rechenregel berücksichtigt werden muss.<sup>64</sup>

Als Wendemarke in der Rechtsprechung des BGH zur Patentierbarkeit von Software kann die Entscheidung „*Logikverifikation*“ vom 13.12.1999 angesehen werden.<sup>65</sup> Gegenstand der Anmeldung war ein Verfahren zur Verifikation hochintegrierter Schaltungen, das einen Zwischenschritt bei der Herstellung von Siliziumchips darstellt. Der Lehre lag die Aufgabe zugrunde, ein Verfahren zu entwickeln, bei dem mit möglichst geringem Speicherplatzbedarf und möglichst kurzer Verarbeitungszeit derartige Prüfungen durchgeführt werden konnten. Bisher erforderten vergleichbare Verfahren einen ungeheueren Speicherplatzbedarf und extrem leistungsfähige Prozessoren. Das der Anmeldung zugrunde liegende Verfahren konnte

---

<sup>63</sup> BGH GRUR 1992, 430ff. – „*Tauchcomputer*“.

<sup>64</sup> BGH GRUR 1992, 430 (432).

<sup>65</sup> BGH GRUR 2000, 498ff. – „*Logikverifikation*“.

künftig im Prinzip auf jedem Universalrechner eingesetzt werden. Das BPatG hatte den technischen Charakter des Verfahrens damit verneint, dass der Kern der Erfindung in einer gedanklich-logischen Anweisung zum Ordnen von Daten bestehe, welche zudem keinen neuen Aufbau oder eine neue Art der Benutzung einer Datenverarbeitungsanlage erfordere.<sup>66</sup> Der BGH hingegen bejahte den technischen Charakter des Verfahrens. Ausschlaggebend sei der Umstand, dass die Anwendung und Programmierung des Verfahrens technische Kenntnisse über den Aufbau elektronischer Schaltungen voraussetze. Da die angemeldete Lehre damit durch eine auf technischen Überlegungen beruhende Erkenntnis und deren Umsetzung geprägt sei, weise sie die für Datenverarbeitungsprogramme erforderliche Technizität auf.<sup>67</sup> Hierbei stützt sich der BGH ausdrücklich auf die Rechtsprechung der Beschwerdekammern des EPA.<sup>68</sup> Unerheblich ist nach Ansicht des BGH, dass auf den Einsatz beherrschbarer Naturkräfte verzichtet wird.<sup>69</sup> Damit kommt der BGH effektiv zu einer Neubewertung seines Technikbegriffs.<sup>70</sup> Auf den Einsatz beherrschbarer Naturkräfte kommt es dann nicht mehr an, wenn eine softwarebezogene Lösung ehemals erforderliche technische Vorgänge substituiert.<sup>71</sup> Durch die Neubewertung seines Technikbegriffs und die Annäherung an die Spruchpraxis des EPA hat der BGH damit die Patentierbarkeit von computerbezogenen Erfindungen erheblich erweitert.

Seine liberalere Haltung gegenüber computerbezogenen Erfindungen setzte der BGH in der Entscheidung „*Sprachanalyseeinrichtung*“ vom 11.05.2000 fort.<sup>72</sup> Anmeldungsgegenstand war ein Computerprogramm zur Ermittlung des Sinngehalts eines in natürlicher Sprache geschriebenen Textes. Die zu lösende Aufgabe bestand darin, die Analyseeinrichtung derart zu programmieren, dass die eingegebenen Texte in der Regel ohne zusätzliche Eingaben des Benutzers analysiert werden konnten, bzw. im Falle eines zusätzlich erforderlichen Hinweises durch den Benutzer, die Anzahl der

---

<sup>66</sup> BPatG GRUR 1998, 656ff. – „CAD/CAM-Verifikationsverfahren“.

<sup>67</sup> BGH GRUR 2000, 498 (500).

<sup>68</sup> Vgl. GRUR Int. 1995, 909 (911) – „SOHEI/Universelles Verwaltungssystem“.

<sup>69</sup> BGH GRUR 2000, 498 (501).

<sup>70</sup> Vgl. zum „nicht-statischen“ Technikbegriff auch *Hössle*, Mitt. 2000, 343ff.

<sup>71</sup> *Busche*, Mitt. 2001, 49 (55); *Ohly*, CR 2001, 809 (812); kritisch hierzu *Nack*, GRUR Int. 2000, 853 (855f.) und *Sedlmaier*, Mitt. 2002, 55 (59).

<sup>72</sup> BGH CR 2000, 500ff. – „Sprachanalyseeinrichtung“.

Auswahlmöglichkeiten wesentlich begrenzt werden konnte. Gegenstand des Patents war jedoch nicht das Computerprogramm als solches, sondern eine auf diese Weise programmtechnisch eingerichtete Datenverarbeitungsanlage. Das BPatG hatte die Patentfähigkeit des Anmeldungsgegenstandes aufgrund fehlender Technizität verneint.<sup>73</sup> Der BGH kam jedoch zum gegenteiligen Ergebnis. Er stellte dabei ausdrücklich klar, dass als Anmeldungsgegenstand weder ein Computerprogramm alleine, noch in Verbindung mit einer Datenverarbeitungsanlage als Verfahren beansprucht wurde, sondern umgekehrt eine auf bestimmte Weise eingerichtete DV-Anlage als Vorrichtung. Einer derartigen Vorrichtung komme der erforderliche technische Charakter *ohne weiteres* zu.<sup>74</sup> Ein Universalrechner sei in jedem Fall als ein technischer Gegenstand anzusehen. Daran ändere auch die Tatsache nichts, dass er in bestimmter Weise programmtechnisch eingerichtet ist. Dies nehme dem Rechner nicht seinen technischen Charakter, sondern füge ihm lediglich weitere Eigenschaften hinzu, auf deren eigenen technischen Charakter es für die Beurteilung der Technizität der Anlage als solche nicht ankomme.<sup>75</sup> Zu Recht wurde im Zuge dieser Entscheidung die Frage gestellt, ob damit nun sämtliche Computerprogramme als patentierbar anzusehen seien, sofern sie nur im Zusammenhang mit einer DV-Anlage als Vorrichtung angemeldet werden.<sup>76</sup> In der Tat ist die unterschiedliche patentrechtliche Beurteilung eines Verfahrens- und eines Vorrichtungsanspruchs wenig überzeugend, da es somit in der Hand des Patentanmelders liegt, durch die bloße Formulierung der Anmeldung über deren Patentierbarkeit zu entscheiden. Jede Anmeldung, welche ein Computerprogramm zum Gegenstand hat, lässt sich auch als Vorrichtungsanspruch formulieren.<sup>77</sup> Dies steht jedoch in krassem Widerspruch zu dem vom BGH bereits in der „*Dispositionsprogramm*“ Entscheidung aufgestellten und mehrfach bestätigten Grundsatz, wonach es bei der Frage der Technizität

---

<sup>73</sup> BPatG GRUR 1999, 411ff. – „Sprachanalyseeinrichtung“.

<sup>74</sup> BGH CR 2000, 500 (501).

<sup>75</sup> BGH CR 2000, 500 (501). Diesem Ansatz hat sich mittlerweile auch die Beschwerdekammer des EPA angeschlossen, vgl. GRUR Int. 2002, 87ff. – „Steuerung eines Pensionssystems“.

<sup>76</sup> Vgl. *Nack*, GRUR Int. 2000, 853 (857).

<sup>77</sup> Bedenken erheben auch *Busche*, Mitt. 2001, 49 (56); *Ohly*, CR 2001, 809 (812).

gerade nicht auf die sprachliche Einkleidung der Anmeldung ankommt.<sup>78</sup>

Bei der jüngsten Entscheidung des BGH zu computerbezogenen Erfindung handelt es sich um die bereits eingangs erwähnte Entscheidung „Suche fehlerhafter Zeichenketten“ vom 17.10.2001.<sup>79</sup> Der BGH nutzt die Gelegenheit um die kaum 12 Monate vorher ergangenen Aussagen in der „Sprachanalyseeinrichtung“ Entscheidung wieder erheblich einzuschränken. Ausweislich des Patentanspruchs ging es um ein Verfahren und ein Computersystem zur Suche und/oder Korrektur fehlerhafter Zeichenketten in einem digital gespeicherten Text. Das BPatG hatte die Patentierbarkeit mit der Begründung verneint, dass es am technischen Charakter fehle und zudem ein Programm als solches im Sinne des § 1 II Nr. 3, III PatG vorläge.<sup>80</sup> Dem wollte der BGH so nicht folgen. Nach umfangreichen Ausführungen zur Bedeutung des Patentierungsverbots des § 1 II Nr. 3, III PatG<sup>81</sup>, stellt der BGH klar, dass eine Lehre nicht schon deshalb als patentierbar angesehen werden kann, weil sie bestimmungsgemäß den Einsatz eines Computers erfordert. Soweit ein Computerprogramm als Vorrichtung angemeldet wird, biete dies keinen Grund, die Frage nach der erforderlichen Patentierfähigkeit allein nach der Kategorie des Anmeldungsgegenstandes und unabhängig davon zu beantworten, was nach der beanspruchten Lehre im Vordergrund stehe.<sup>82</sup> Dies gelte auch dann, wenn ein Computerprogramm in einer auf einem herkömmlichen Datenträger (z.B. Diskette, CD-ROM) gespeicherten Form angemeldet wird. Auch in diesem Fall komme es darauf an, ob die prägenden Anweisungen des Verfahrens der Lösung eines konkreten technischen Problems dienen.<sup>83</sup> Ob dies im vorliegenden Fall gegeben war, konnte der BGH mangels hinreichender tatsächlicher Feststellungen nicht abschließend beurteilen und verwies die Sache zurück an das BPatG.

---

<sup>78</sup> BGH GRUR 1977, 96 (97) – „Dispositionsprogramm“; BGH GRUR 1978, 102 – „Prüfverfahren“; BGH GRUR 1981, 39 (40) – „Walzstabteilung“; BGH GRUR 1986, 531 (533) – „Flugkostenminimierung“.

<sup>79</sup> BGH CR 2002, 88ff. – „Suche fehlerhafter Zeichenketten“.

<sup>80</sup> BPatG CR 2001, 155 (156ff.) – „Fehlerhafte Zeichenketten“.

<sup>81</sup> Vgl. dazu bereits oben S. 8f.

<sup>82</sup> BGH CR 2002, 88 (91).

<sup>83</sup> BGH CR 2002, 88 (91).

### **(b) Neuere Rechtsprechung des Bundespatentgerichts**

In der Entscheidung „*Verfahren zur Erstellung von Mailing-Adresslisten*“ vom 13.12.2001 ging es um ein Computerprogramm zur automatischen Erstellung und Aktualisierung von Adresslisten für Mailingaktionen.<sup>84</sup> Das BPatG verneinte die Patentierbarkeit des Verfahrens unter Hinweis auf die fehlende Technizität. Die Erstellung und Aktualisierung von Mailinglisten beruhe allein auf der Kenntnis der organisatorischen Zusammenhänge und Zielsetzungen für die Durchführung von Mailingaktionen, ohne dass technische Erkenntnisse erforderlich wären.<sup>85</sup> Der technischer Charakter des Verfahrens ergebe sich auch nicht daraus, dass das Programm bestimmungsgemäß auf einem Rechner und damit auf einem technischen Gerät zum Einsatz komme. Unter Berufung auf die Entscheidung „*Suche fehlerhafter Zeichenketten*“ des BGH, stellt das BPatG vielmehr fest, dass der Einsatz einer technischen Datenverarbeitungsanlage für sich genommen nicht ausreicht, um ein Verfahren als technisch anzusehen, vielmehr müssten auch hier die prägenden Anweisungen der Lösung eines konkreten technischen Problems dienen.<sup>86</sup>

In der am 09.04.2002 ergangenen Entscheidung „*Keine Patentierbarkeit geschäftlicher Tätigkeit*“ musste sich das BPatG mit einem Verfahren und einer Vorrichtung für die effiziente computergestützte Steuerung von Geschäftsprozessen und Verfahrensabläufen befassen.<sup>87</sup> Gegenstand der Anmeldung war ein Computerprogramm, welches den Typ eines Dokuments und relevante inhaltliche Angaben automatisch ermitteln und verarbeiten konnte und damit die Tätigkeit eines Menschen bei der Abwicklung geschäftlicher Abläufe ersetzen konnte. Das BPatG kommt in strenger Anwendung der in der „*Suche fehlerhafter Zeichenketten*“ Entscheidung aufgestellten Grundsätze zu dem Ergebnis, dass das Verfahren nicht als ein technisches Programm anzusehen ist und auch keine Eigenheit aufweist, die unter Berücksichtigung der Zielsetzung des PatG eine Patentierbarkeit rechtfertigt.<sup>88</sup> Hinsichtlich des Vorrichtungsanspruchs führt das BPatG aus, dass auch hier im

---

<sup>84</sup> BPatG CR 2002, 248f. – „*Verfahren zur Erstellung von Mailing-Adresslisten*“.

<sup>85</sup> BPatG CR 2002, 248.

<sup>86</sup> BPatG CR 2002, 248 (249).

<sup>87</sup> BPatG CR 2002, 716ff. – „*Keine Patentierbarkeit geschäftlicher Tätigkeit*“.

<sup>88</sup> BPatG CR 2002, 716 (717).

Vordergrund des formal auf ein Computersystem gerichteten Anspruchs, die Anweisungen des Verfahrens stehen, so dass keine andere Bewertung gerechtfertigt sei.<sup>89</sup>

In der darauf folgenden Entscheidung „*Kein Patentanspruch für Cyber-Cash-Verfahren*“ musste sich das BPatG mit einem Verfahren zum elektronischen Zahlungsverkehr im Internet beschäftigen.<sup>90</sup> Das BPatG bejahte die Technizität des Verfahrens, weil sämtliche Verfahrensschritte automatisch und auf elektronischen Wege erfolgten. Eine Gesamtbetrachtung der beanspruchten Lehre ergebe daher, dass die prägenden Anweisungen der Lösung eines technischen Problems dienen.<sup>91</sup> Daran ändere auch die Tatsache nichts, dass auch geschäftliche Inhalte im Vordergrund des beanspruchten Verfahrens stünden. Die Patentierbarkeit des Verfahrens verneinte das BPatG jedoch wegen der fehlenden Erfindungshöhe.<sup>92</sup>

In seiner neuesten Entscheidung „*Rechnergestützte Sprachanalyse*“ vom 12.11.2002 musste sich das BPatG mit einem Verfahren zur rechnergestützten Sprachanalyse, bei dem einer sprachlichen Äußerung eine syntaktische Struktur zugeordnet wird, befassen.<sup>93</sup> Die zu lösende Aufgabe der beanspruchten Lehre bestand darin, den nach der grammatikalischen Zerlegung eines Satzes möglichen syntaktischen Strukturen anhand der sprachlichen Eigenschaften des Satzes (sog. erweiterter Kontext) Wahrscheinlichkeiten für ihre Richtigkeit zuzuordnen. Das BPatG sah die prägenden Anweisungen dieser Lehre nicht auf dem Gebiet der Technik, sondern auf dem Gebiet der Sprache.<sup>94</sup> Nach Ansicht des BPatG wies die beanspruchte Lehre auch keine Eigenheit auf, die unter Berücksichtigung der Zielsetzung des PatG eine Patentierbarkeit rechtfertigt. Insbesondere könne nicht etwa der von der Anmelderin angeführte hohe Entwicklungsaufwand als Rechtfertigung für eine Patentierbarkeit angesehen werden.<sup>95</sup>

---

<sup>89</sup> BPatG CR 2002, 716 (718).

<sup>90</sup> BPatG CR 2002, 559ff. – „Kein Patentanspruch für Cyber-Cash-Verfahren“.

<sup>91</sup> BPatG CR 2002, 559.

<sup>92</sup> Vgl. dazu unten III.1.c.

<sup>93</sup> BPatG Mitt. 2003, 207ff. – „Rechnergestützte Sprachanalyse“.

<sup>94</sup> BPatG Mitt. 2003, 207 (209).

<sup>95</sup> BPatG Mitt. 2003, 207 (209).

### (c) Ergebnis

Die bisherige Rechtsprechung des BGH zur Patentierbarkeit von computerbezogenen Erfindungen wurde zum Teil harscher Kritik ausgesetzt. So wurde ihr vorgeworfen, sie rufe „eine signifikante Rechtsunsicherheit“<sup>96</sup> hervor, sie stelle ein „dogmatisches Chaos“<sup>97</sup> dar, sie offenbare „ein Kaleidoskop – häufig zudem unvereinbarer – Norminterpretationen“<sup>98</sup> oder sie sei von einer „Kette mehr oder minder unglücklicher Formulierungen geprägt“<sup>99</sup>. Rückblickend können vor allem die noch zur Kerntheorie ergangenen Entscheidungen kaum überzeugen. Die Kerntheorie lieferte kein greifbares Kriterium um zwischen technischen und untechnischen Programmen zu unterscheiden. Der Kern der Lehre wurde vom BGH und vom BPatG eher willkürlich festgelegt, so dass weder in der Entscheidungspraxis des BGH, noch in der Entscheidungspraxis des BPatG eine einheitliche Linie zu erkennen war. Zudem vermischte der BGH bei der Kerntheorie Merkmale der erfinderischen Tätigkeit mit solchen der Technizität.<sup>100</sup> Dabei handelt es sich aber um zwei getrennt zu betrachtende Prüfungspunkte. Mit dem Wechsel zur Gesamtbetrachtungslehre konnte die Bewertung des technischen Charakters von Computerprogrammen auf ein verlässlicheres, wenn auch nicht vollkommen widerspruchsfreies Fundament gestellt werden. Auch der in der Entscheidung „*Logikverifikation*“ eingeleitete Verzicht auf das Kriterium des Einsatzes beherrschbarer Naturkräfte ist zu begrüßen, da insoweit der Entwicklung auf dem Gebiet der Informatik Rechnung getragen wird, welche zunehmend ehemals maschinelle Prozesse durch den Einsatz von Computerprogrammen ersetzt. Als bedenklich sind hingegen die Ausführungen des BGH in der Entscheidung „*Sprachanalyseeinrichtung*“ einzustufen. Die Ausführungen zur Technizität eines Vorrichtungsanspruchs führten zu erheblichen Missverständnissen. Auch wenn der BGH in der darauf folgenden Entscheidung „*Suche fehlerhafter Zeichenketten*“ versuchte, diese ausufernden Feststellungen wieder einzuschränken, ist das Verhältnis von Verfahrens- zu Vorrichtungsansprüchen nach wie vor unklar. Inwiefern es dem BGH in derselben Entscheidung

---

<sup>96</sup> *Horns*, GRUR 2001, 1 (10).

<sup>97</sup> *Nack*, GRUR Int. 2000, 853 (856).

<sup>98</sup> *Busche*, Mitt. 2001, 50 (56).

<sup>99</sup> *Schmidtchen*, Mitt. 1999, 281 (284).

<sup>100</sup> So auch *Busche*, Mitt. 2001, 50 (51); *Nack*, GRUR Int. 2000, 853 (854); *Ohly*, CR 2001, 809 (811).

gelingen ist, das Patentierungsverbot für Computerprogramme „als solche“ gemäß § 1 II Nr. 3, III PatG einer überzeugenden Definition zuzuführen, ist fraglich. Der BGH zäumt das Pferd von hinten auf, wenn er ausführt, dass es sich nicht um ein Computerprogramm „als solches“ handelt, wenn die prägenden Anweisungen der beanspruchten Lehre der Lösung eines konkreten technischen Problems dienen.<sup>101</sup> Im Umkehrschluss lässt sich also folgern, dass unter ein Computerprogramm „als solches“ jedes Programm fällt, was nicht der Lösung eines technischen Problems dient. Die jedem Programm dabei innewohnende „Grundtechnizität“ reicht dafür nach Ansicht des BGH jedenfalls nicht aus.<sup>102</sup> Damit geht der BGH konform mit der jüngeren Rechtsprechung der Beschwerdekammern des EPA, welche auf das Erfordernis eines „weiteren technischen Effekts“ abstellen.<sup>103</sup> Fraglich ist freilich, wann ein Computerprogramm der Lösung eines konkreten technischen Problems dient. Nach Ansicht des BGH ist dies jedenfalls bei Ansprüchen, die zur Lösung eines Problems auf den herkömmlichen Gebieten der Technik, also der Ingenieurwissenschaften, der Physik, der Chemie oder der Biologie, die Abarbeitung bestimmter Verfahrensschritte durch einen Computer vorschlagen, unproblematisch zu bejahen.<sup>104</sup> Ansonsten bedarf es einer speziellen Prüfung dahingehend, ob die zugrunde liegende Lehre sich gerade durch eine Eigenheit auszeichnet, die unter Berücksichtigung der Zielsetzung patentrechtlichen Schutzes eine Patentierbarkeit rechtfertigt.<sup>105</sup> Diese Ausführungen des BGH lassen sich anhand seiner bisherigen Rechtsprechung durch die Bildung von Fallgruppen mit Leben füllen:

- Zweifellos patentierbar sind danach Programme aus dem Bereich der Steuerungs- und Regeltechnik, welche regelnd in mechanische Prozesse eingreifen.
- Patentierbar sind aber auch Programme, die auf technischen Überlegungen beruhen und darauf abzielen, herkömmliche technische Vorgänge durch Rechenoperationen zu substituieren.<sup>106</sup>

---

<sup>101</sup> BGH CR 2002, 88 (90) – „Suche fehlerhafter Zeichenketten“.

<sup>102</sup> BGH CR 2002, 88 (90); ebenso BPatG CR 2002, 716 (717).

<sup>103</sup> Vgl. GRUR Int. 1999, 1053ff. – „Computerprogrammprodukt/IBM I“; Entscheidung der Technischen Beschwerdekammer 3.5.1 v. 04.02.1999 ( T 0935/97).

<sup>104</sup> BGH CR 2002, 88 (90).

<sup>105</sup> BGH CR 2002, 88 (90).

<sup>106</sup> *Busche*, Mitt. 2001, 49 (56); *Ohly*, CR 2001, 809 (815).

Hierzu sind z.B. Computerprogramme aus dem Bereich CAD/CAM zu rechnen.

- Patentierbar sind ferner Programme, die die Funktionsfähigkeit einer Datenverarbeitungsanlage als solche betreffen und damit das unmittelbare Zusammenwirken ihrer Elemente ermöglichen. Dazu dürften insbesondere Betriebssysteme und deren Komponenten zu rechnen sein.<sup>107</sup>

Nicht als patentierbare Programme anzusehen sind jedoch:

- Programme, deren prägende Anweisungen nicht der Lösung eines technischen Problems dienen. Hierunter sind nach der bisherigen Rechtsprechung vor allem reine Anwendungsprogramme wie Textverarbeitungen, Tabellenkalkulationen oder Programme mit betriebswirtschaftlichem Hintergrund einzuordnen.
- Dies muss auch dann gelten, wenn sie in Form eines Vorrichtungsanspruchs angemeldet werden.

Betrachtet man die neuere Rechtsprechung des BPatG, so ergibt sich ein weitgehend einheitliches Bild. Das BPatG hat die vom BGH in der Entscheidung „*Suche fehlerhafter Zeichenketten*“ dargelegten Leitlinien übernommen und konsequent angewendet. Das BPatG untersuchte in allen vier Entscheidungen, ob die prägenden Anweisungen der beanspruchten Lehre der Lösung eines technischen Problems dienen und setzte sich in überzeugender Art und Weise mit der Frage auseinander, ob im konkreten Fall eine Eigenheit vorlag, die unter Berücksichtigung der Zielsetzung patentrechtlichen Schutzes eine Patentierbarkeit der beanspruchten Lehre rechtfertigte. Zusammenfassend kann daher festgehalten werden, dass die neuere Rechtsprechung des BGH und des BPatG jedenfalls dem ersten Anschein nach zu einer konsequenteren und einheitlicheren Behandlung von computerbezogenen Erfindungen geführt hat. Es bleibt abzuwarten, inwiefern sich dieser Trend fortsetzt, oder ob die gefundene Linie – wie es schon häufig der Fall war – durch neue Entscheidungen wieder aufgelöst wird.

---

<sup>107</sup> Jaeger/Metzger, OSS, S. 110.

## (2) Meinungsstand in der Literatur

Die Frage nach der Patentierbarkeit von Computerprogrammen wurde auch in der juristischen Literatur kontrovers behandelt. Im Gegensatz zur Rechtsprechung, wurde sich hier aber schon frühzeitig mit dem Patentierungsverbot des § 1 II Nr. 3, III PatG auseinandergesetzt. Die überwiegende Ansicht geht zunächst davon aus, dass Computerprogramme generell technischen Charakter besitzen.<sup>108</sup> Während vereinzelte Stimmen<sup>109</sup> die „als solche“ Formel für eine weitgehend inhaltsleere und überflüssige Regelung halten, versteht die wohl h.M. unter Computerprogrammen „als solche“ alle nicht-technischen Programme.<sup>110</sup> Auch die Prüfungsrichtlinien des EPA<sup>111</sup> und des DPA<sup>112</sup> unterscheiden bei Art. 52 II c), III EPÜ bzw. § 1 II Nr. 3, III PatG zwischen technischen und nicht technischen Programmen. Teilweise wird diese Ansicht dahingehend modifiziert, dass nur solche Programme als patentierbar anzusehen sind, deren technischer Charakter in Eigenschaften und Funktionen begründet ist, die außerhalb der grundlegenden gemeinsamen Eigenschaften aller Programme liegen.<sup>113</sup> Dieser Auffassung ist jedoch entgegenzuhalten, dass sich dieses Erfordernis ohnehin aus den §§ 3 und 4 PatG (Neuheit und erfinderische Tätigkeit) ergibt. *Winischhofer* wiederum stellt darauf ab, ob das angemeldete Programm einen technischen Effekt in der physikalischen Welt hervorruft.<sup>114</sup> Fehlt es an einem solche „Effekt in der Außenwelt“, liegt ein Computerprogramm als solches vor.<sup>115</sup> Es fragt sich aber, ob damit nicht an einem Technikbegriff festgehalten wird, der die Entwicklungen auf dem Gebiet der Informatik nicht mehr angemessen berücksichtigt.

---

<sup>108</sup> Vgl. *Anders*, GRUR 1990, 498 (500); *Betten/Esslinger*, CR 2000, 18 (19); *Ohly*, CR 2001, 809 (813); *Klopmeier*, Mitt. 2002, 65 (70); *Melullis*, FS-Erdmann, S. 408; ders.; GRUR 1998, 843 (851); *Horns*, GRUR 2001, 1 (13); *Nack*, GRUR Int. 2000, 853 (858); *Schiurma*, GRUR Int. 1998, 852 (854); *Schmidtchen*, Mitt. 1999, 281 (284); a.A. *Metzger*, CR 2003, 313 (315); *Ochmann* in: FS-Nirk, S. 771; *Schölch*, GRUR 2001, 16 (18).

<sup>109</sup> v. *Hellfeld*, GRUR 1989, 471 (476); *Horns*, GRUR 2001, 1 (8).

<sup>110</sup> *Busche*, Mitt. 2000, 164 (171); *Busse-Keukenschrijver*, PatG, § 1, Rn. 37; *Goebel*, FS-Nirk, S. 372; *Kraßer* in: *Lehmann*, Rechtsschutz für Computerprogramme, S. 270, Rn. 104; *Schar*, Mitt. 1998, 322 (338); *Schulte*, PatG, § 1, Rn. 60.

<sup>111</sup> Vgl. Teil C, Abschnitt IV, 2.3 der Prüfungsrichtlinien, abrufbar unter: [http://www.european-patent-office.org/legal/gui\\_lines/pdf/index.html](http://www.european-patent-office.org/legal/gui_lines/pdf/index.html).

<sup>112</sup> Vgl. 4.3.3. DPA-Richtlinien vom 02. Juni 1995, abrufbar unter: <http://transpatent.com/gesetze/prl.html>.

<sup>113</sup> *Engel*, GRUR 1993, 194 (197).

<sup>114</sup> *Winischhofer*, Computersoftware und Patentrecht, S. 99ff.; ähnlich auch *Schar*, Mitt. 1998, 322 (334).

<sup>115</sup> *Winischhofer*, Computersoftware und Patentrecht, S. 100.

Andere Autoren sehen in dem Kriterium der Technizität kein für § 1 II Nr. 3, III PatG geeignetes Angrenzungskriterium. *Klopmeier* stellt daher darauf ab, ob das beanspruchte Programm der Lösung eines (nicht unbedingt technischen) Problems dient.<sup>116</sup> Nur Programme, die nicht der Lösung eines Problems dienen, seien als Programme „als solche“ anzusehen.<sup>117</sup> Hiergegen ist jedoch einzuwenden, dass im Prinzip jedes Programm der Lösung eines Problems dient, so dass fraglich ist, wie hier zwischen Programmen „als solchen“ und anderen patentierfähigen Programmen zu unterscheiden ist. Von anderen Autoren wird zur Abgrenzung zwischen Programmen „als solchen“ und anderen Programmen auf die Schnittstelle zwischen Urheber- und Patentrecht abgestellt. Nach *Tauchert* ist der reine urheberrechtlich geschützte Quellcode als Programm „als solches“ anzusehen.<sup>118</sup> Die „runtime version“ eines Programms hingegen sei patentfähig. Auch wenn diese Ansicht zunächst eine klare Interpretation des § 1 II Nr. 3, III PatG zu liefern scheint, überzeugt sie bei näherer Betrachtung nicht. Die von *Tauchert* aufgestellte Behauptung, wonach nur der Quellcode eines Programms urheberrechtlichen Schutz genießt, wird durch einen Blick auf § 69a I und II UrhG widerlegt. So heißt es dort, dass der Schutz für Computerprogramme „in jeder Gestalt“ und „für alle Ausdrucksformen“ gewährt wird. Dazu zählt aber auch die kompilierte „runtime version“ eines Programms. Nach *Melullis* wiederum ist das gedanklich-logische Konzept, welches der durch die Software zu lösenden Aufgabe zugrunde liegt, d.h. der eigentliche, von einer technischen Funktion noch freie, Programminhalt als Programm „als solches“ anzusehen.<sup>119</sup> Insgesamt zeigt die Vielfalt an Interpretationsvorschlägen in der Literatur, wie schwer es fällt, die „als solche“ Formel einer dogmatisch überzeugenden und praktisch handhabbaren Bedeutung zuzuführen.

### (3) Eigene Stellungnahme

Wie der BGH in seiner Entscheidung *„Suche fehlerhafter Zeichenketten“* richtigerweise feststellt, muss die Bedeutung des Patentierungsverbots für Computerprogramme gemäß § 1 II Nr. 3, III

---

<sup>116</sup> *Klopmeier*, Mitt. 2002, 65 (69f.).

<sup>117</sup> *Klopmeier*, Mitt. 2002, 65 (69), nennt hier als Beispiel Computerspiele.

<sup>118</sup> *Tauchert*, Mitt. 1999, 248 (250); ders. JurPC Web-Dok. 28/2002, Abs. 38.

<sup>119</sup> *Melullis*, GRUR 1998, 843 (851); ähnlich auch *Anders*, GRUR 1990, 498 (499); *Ohly*, CR 2001, 809 (813).

PatG, wie bei jeder anderen Rechtsvorschrift auch, durch Auslegung anhand der üblichen Auslegungskriterien ermittelt werden.

Ausgangspunkt einer jeden Auslegung ist der Wortlaut einer gesetzlichen Regelung. Aus dem Wortlaut des § 1 II PatG ergibt sich zunächst relativ eindeutig, dass die dort in den Nr. 1 – 4 aufgeführten Gegenstände und Tätigkeiten nicht als patentierfähige Erfindungen im Sinne des § 1 I PatG anzusehen sind. Der Katalog des § 1 II PatG ist dabei, wie sich aus dem Wort „insbesondere“ ergibt, nicht abschließend, sondern nur exemplarisch zu verstehen. Die Grundregel des § 1 II PatG wird jedoch durch Abs. 3 wiederum dahingehend modifiziert, dass das Patentierungsverbot nur insoweit gilt, als für die genannten Gegenstände oder Tätigkeiten „als solche“ Schutz begehrt wird. Die Einschränkung „als solche“ könnte man auch mit der Formulierung „so wie sie in Abs. 2 ihrer Art und Beschaffenheit nach charakterisiert sind“ oder „für sich allein betrachtet“<sup>120</sup> ersetzen. Eine Anmeldung, welche von der Patentierung ausgeschlossen ist, muss sich also auf etwas beziehen, was seiner Art und Beschaffenheit nach, also vom Phänotyp her gesehen, durch einen der in Abs. 2 genannten Begriffe abschließend charakterisiert wird.<sup>121</sup> Es kommt also für die Auslegung der Einschränkung „als solche“ entscheidend darauf an, was unter einem Programm für eine Datenverarbeitungsanlage im Sinne des § 1 II PatG zu verstehen ist. Hierbei stellt sich jedoch das Problem, dass dem Begriff „Programm für eine Datenverarbeitungsanlage“ keineswegs ein eindeutiger Wortsinn entnommen werden kann. So kann darunter z.B. der in einer Programmiersprache verfasste Text zu verstehen sein. Andererseits kann darunter aber auch die kompilierte, also ausführbare und auf einem Datenträger abgespeicherte Form zu verstehen sein.<sup>122</sup> Schließlich wäre es auch denkbar, darunter nur ein Programm zu verstehen, welches gerade auf einem Computer ausgeführt wird. Es ist nicht erkennbar, dass eine dieser Formen im allgemeinen Sprachgebrauch als Computerprogramm „als solches“ bezeichnet wird. Nur anhand des Wortlauts des § 1 II Nr. 3, III PatG ist es daher nicht möglich die Bedeutung der Einschränkung „als solche“ zu erfassen.

---

<sup>120</sup> *Benkhard-Bruchhausen*, PatG, § 1, Rn. 95; *Engel*, GRUR 1993, 194 (195).

<sup>121</sup> *König*, GRUR 2001, 577 (582); *Nack*, GRUR Int. 2000, 853 (854); *Pierson*, Der Schutz der Programme für die Datenverarbeitung, S. 169.

<sup>122</sup> Vgl. *BPatG*, CR 2001, 155 (156).

Es fragt sich, ob sich aus dem systematischen Zusammenhang der verschiedenen in § 1 II PatG genannten Gruppen und einer Auslegung nach Sinn und Zweck des § 1 II, III PatG eine überzeugende Interpretation für das Patentierungsverbot für Computerprogramme „als solche“ finden lässt. Bemerkenswert ist dabei zunächst, dass sich die Einschränkung „als solche“ auf sämtliche in Abs. 2 genannten Gegenstände und Tätigkeiten bezieht. Da es sich bei den in Abs. 2 genannten Gegenständen und Tätigkeiten um teilweise recht unterschiedliche Fallgruppen handelt, liegt der Schluss nahe, dass die Einschränkung „als solche“ nicht bei allen Fallgruppen die gleiche Reichweite haben muss.<sup>123</sup> Die in § 1 II Nr. 1 PatG aufgeführten Tätigkeiten (Entdeckungen, wissenschaftliche Theorien und mathematische Formeln) sind für sich genommen nicht technisch, weil sie nicht der Erreichung eines kausal übersehbaren Erfolges dienen. Patentierbar ist aber eine Erfindung, welche auf einer Entdeckung, einer wissenschaftlich Theorie oder einer mathematischen Formel beruht, oder eine solche zur Erreichung eines konkreten Erfolgs verwendet. Auch bei der in § 1 II Nr. 2 PatG genannten Fallgruppe der ästhetischen Formschöpfungen ist die „als solche“ Einschränkung dahingehend zu verstehen, dass eine ästhetische Formschöpfung „als solche“ niemals eine patentierfähige Erfindung darstellen kann, weil sie in keiner Weise die Anforderungen an den Technikbegriff erfüllt. Eine Erfindung kann aber sehr wohl auch eine ästhetische Formschöpfung darstellen.<sup>124</sup> Auch die vierte Fallgruppe des § 1 II PatG (Wiedergabe von Informationen) stellt für sich genommen keine technische Erfindung dar. Eine Erfindung kann aber sehr wohl auch die Wiedergabe von Informationen beinhalten. Den in § 1 II Nr. 3 PatG genannten Plänen, Regeln und Verfahren für gedankliche Tätigkeiten, für Spiele und für geschäftliche Tätigkeiten ist gemein, dass es sich jeweils um Anweisungen an den menschlichen Geist handelt.<sup>125</sup> Zwar können sie der Erreichung eines kausal übersehbaren Erfolges dienen, auch ihnen fehlt aber der technische Charakter, weil sie nicht die planmäßige Benutzung oder Substitution beherrschbarer Naturkräfte *außerhalb der menschlichen Verstandeskraft* zum Gegenstand haben. Sämtlichen in § 1 II PatG

---

<sup>123</sup> So auch *Engel*, GRUR 1993, 194 (196).

<sup>124</sup> Hier ist nochmals an das bereits oben dargestellte Nebeneinander von urheberrechtlichem- und patentrechtlichem Schutz zu erinnern.

<sup>125</sup> *Benkhard-Bruchhausen*, PatG, § 1, Rn. 101; *Kraßer* in: *Lehmann*, Rechtsschutz von Computerprogrammen, S. 235, Rn. 38.

genannten Gegenständen und Tätigkeiten (unter Außerachtlassung von Computerprogrammen) fehlt es demnach an einem technischen Charakter.<sup>126</sup> Sinn und Zweck des § 1 II PatG ist es also, dem Rechtsanwender durch eine beispielhafte Auflistung von nichttechnischen Gegenständen, die Auslegung des Erfindungsbegriffs zu erleichtern. Übergeordnetes Ziel ist dabei, angesichts des protektionistischen Charakters des Patentrechts, den Patentschutz auf solche Erfindungen zu begrenzen, die dem technischen Fortschritt dienen und nicht zu einer Monopolisierung des menschlichen Denkens führen. Gleichzeitig soll durch Abs. 3 aber gewährleistet werden, dass durch die Auflistung in § 1 II PatG nicht jede Erfindung vom Patentschutz ausgeschlossen wird, die sich *auch* einer der genannten Gegenstände oder Tätigkeiten bedient bzw. diese inkorporiert. Aus der Gesetzessystematik und durch Auslegung nach Sinn und Zweck der Regelung lässt sich damit eine überzeugende Interpretation des Patentierungsverbots für alle anderen in § 1 II PatG genannten Gegenstände und Tätigkeiten finden.

Für Computerprogramme gestaltet sich diese Auslegung jedoch ungleich schwieriger. Es kann zunächst nicht bezweifelt werden, dass Computerprogramme technischen Charakter besitzen.<sup>127</sup> Dies ergibt sich daraus, dass jedes Computerprogramm als eine geordnete Anweisung zum Betrieb eines Universalrechners anzusehen ist. Ein Universalrechner wiederum ist zweifellos ein technischer Gegenstand. Da jedes Computerprogramm insofern als eine Anweisung zur Nutzung eines technischen Geräts anzusehen ist, kann ihm der technische Charakter nicht ernsthaft abgesprochen werden.<sup>128</sup> Bestätigt wird dieses Ergebnis durch die Feststellung, dass jedes Computerprogramm auch als ein festverdrahteter Spezialrechner und damit als Hardware zu realisieren ist.<sup>129</sup> Bei der Frage der Technizität kommt es auch nicht darauf an, in welcher Form ein Computerprogramm vorliegt. Der Zweck auf einem Universalrechner ausgeführt zu werden ist jedem Computerprogramm inhärent, egal ob es als Quellcode oder in kompilierter Form, auf einem Datenträger oder auf Papier vorliegt. Insofern ist *v. Hellfeld* dahingehend

---

<sup>126</sup> So auch *Mellullis*, GRUR 1998, 843 (845).

<sup>127</sup> Vgl. schon oben Fn. 108.

<sup>128</sup> So schon BGH GRUR 1977, 96 (98) „Dispositionsprogramm“.

<sup>129</sup> *Engel*, GRUR 1993, 194 (196); *Mellullis*, GRUR 1998, 843 (852); a.A. *Kraßer* in: *Lehmann*, Rechtsschutz für Computerprogramme, S. 271, Rn. 106.

zuzustimmen, dass es ein Computerprogramm „als solches“ gar nicht gibt.<sup>130</sup> Jedes Computerprogramm hat einen Zweck, der außerhalb seines Seins als Programm liegt. Jedes Computerprogramm besitzt stets sowohl Form als auch Funktion. Insofern unterscheiden sich Computerprogramme auch in eklatanter Weise von den restlichen in § 1 II Nr. 1 – 4 PatG aufgeführten Gegenständen und Tätigkeiten. Denn nur Computerprogramme weisen diese spezielle Dualität auf. Während es daher ohne weiteres vorstellbar ist, was unter einer Entdeckung, einer wissenschaftlichen Theorie, einer mathematischen Formel, etc. „als solche“ zu verstehen ist, läuft die „als solche“ Formel bei Computerprogrammen in dieser Hinsicht leer. Programme für Datenverarbeitungsanlagen passen daher im Grunde genommen nicht zu den restlichen in § 1 II PatG aufgeführten Gegenständen und Tätigkeiten. Dies wird auch durch einen Blick auf die Entstehungsgeschichte von Art. 52 II EPÜ bestätigt, welcher der Regelung des § 1 II PatG zugrunde liegt. Der Ausschluss von Computerprogrammen wurde erst relativ spät (Entwurf 1971) und zwar in Anpassung an die Patent Cooperation Treaty (PCT)<sup>131</sup> in das EPÜ eingefügt.<sup>132</sup> Der Ausschluss für Computerprogramme in der PCT wiederum beruhte auf rein pragmatischen Überlegungen, nämlich den mangelnden Recherchemöglichkeiten der Patentämter bei Softwareerfindungen Rechnung zu tragen.<sup>133</sup> Die „als solche“ Formel bezog sich also von Anfang an nur auf die anderen in Art. 52 II EPÜ bzw. § 1 II PatG genannten Gegenstände und Tätigkeiten. Computerprogramme wurden erst später quasi als Fremdkörper in den Katalog des Art. 52 II EPÜ respektive § 1 II PatG eingefügt.

Geht man nun aber davon aus, dass jedes Computerprogramm technischen Charakter besitzt, so wäre es widersinnig die Einschränkung „als solche“ lediglich als eine Wiederholung des Technizitätskriteriums zu verstehen. Da das Technizitätskriterium nämlich ohnehin unabdingbare Voraussetzung des Erfindungsbegriffs

---

<sup>130</sup> v. *Hellfeld*, GRUR 1989, 471 (476); zustimmend *van Raden*, GRUR 1995, 451 (456).

<sup>131</sup> Vertrag über die internationale Zusammenarbeit auf dem Gebiet des Patentwesens vom 19.06.1970; Text abrufbar unter: <http://www.wipo.org/pct/de/texts/pdf/pct.pdf>.

<sup>132</sup> Vgl. dazu *Melullis*, FS-*Erdmann*, S. 407; *Schar*, Mitt. 1998, 322 (333); *Schmidtchen*, Mitt. 1999, 281 (282).

<sup>133</sup> Vgl. *Engel*, GRUR 1993, 194 (196); *Schmidtchen*, Mitt. 1999, 281 (282); *Schar*, Mitt. 1998, 322 (333).

ist, wäre die „als solche“ Formel vollkommen überflüssig.<sup>134</sup> Andererseits kann die jedem Computerprogramm innewohnende „Grundtechnizität“ nicht ausreichen, um jedes Programm als patentfähig anzusehen, denn dann hätte das Patentierungsverbot des § 1 II PatG überhaupt keinen Anwendungsbereich mehr. Stellt man nun wieder auf Sinn und Zweck der Regelung des § 1 II, III PatG ab, nämlich den Patentschutz auf solche Erfindungen zu begrenzen, die dem technischen Fortschritt dienen und nicht zu einer Monopolisierung des menschlichen Denkens führen, so muss dies auch (und erst recht) für Computerprogramme gelten. Dies gilt umso mehr, als sich nahezu sämtliche der anderen in § 1 II PatG genannten Gegenstände und Tätigkeiten auch mittels eines Computerprogramms umsetzen lassen. Aufgrund des technischen Charakters von Computerprogrammen muss die Einschränkung „als solche“ allerdings etwas modifiziert werden, um diesem Zweck gerecht zu werden. Ein Computerprogramm „als solches“ ist damit grundsätzlich jedes Computerprogramm, sofern es nicht zur Lösung eines technischen Problems beiträgt, welches über die bloße Implementierung als Computerprogramm hinausgeht. Dies setzt entgegen der Ansicht von *Winischhofer*<sup>135</sup> nicht voraus, dass ein Effekt in der physikalischen Welt hervorgerufen wird. Andererseits genügt auch nicht die bloße Umsetzung eines bekannten oder unbekanntes Verfahrens auf einem Computer. In Einklang mit den übergeordneten Zielen des Patentschutzes können nur konkrete Lösungen technischer Probleme eine Patentierbarkeit rechtfertigen.

Abzugrenzen ist an dieser Stelle auch zwischen einem Computerprogramm und dem, ihm zugrunde liegenden Algorithmus. Ein Algorithmus ist ein abstrakter Lösungsweg, bestehend aus einer endlichen Anzahl von Arbeitsregeln, deren schrittweise Befolgung der Lösung einer bestimmten Aufgabe dient.<sup>136</sup> Er kann auf unterschiedlichste Art und Weise dargestellt werden, z.B. mittels mathematischer Symbole, als Flussdiagramm oder in natürlicher Sprache. Als eine durch reine Logik definierte, auf abstrakten

---

<sup>134</sup> So auch *König*, GRUR 2001, 577 (580); *Melullis*, FS-Erdmann, S. 408; *van Raden*, GRUR 1995, 451 (456).

<sup>135</sup> *Winischhofer*, Computersoftware und Patentschutz, S. 100.

<sup>136</sup> *Horns*, GRUR 2001, 1 (7); *Kolle*, GRUR 1977, 58 (66); *van Raden*, GRUR 1995, 451 (455); *Pierson*, Der Schutz der Programme für die Datenverarbeitung, S. 22f; *Marly*, Computersoftware in der EU, S. 112f.

Beschreibungsebene angesiedelte Arbeitsanweisung, fehlt es dem Algorithmus an jeglichem Bezug zu einer konkreten realen Implementierung. Als solcher stellt ein Algorithmus daher stets eine Regel oder ein Verfahren für eine gedankliche Tätigkeit dar und ist daher gemäß § 1 II Nr. 3, III PatG nicht patentierbar.<sup>137</sup> Aufgrund seiner abstrakten Natur, mangelt es einem Algorithmus folglich am Merkmal der Technizität. Erst durch seine Umsetzung in ein Computerprogramm zur Lösung eines konkreten technischen Problems entsteht eine patentierfähige Erfindung.<sup>138</sup>

Dieses Verständnis eines Computerprogramms „als solches“ steht in Einklang mit Sinn und Zweck des Patentschutzes und steht auch nicht im Widerspruch zum Wortlaut des § 1 II Nr. 3, III PatG, der insoweit die Grenze der zulässigen Auslegung bildet. Ferner geht diese Auslegung auch konform mit der Intention des historischen Gesetzgebers, der angesichts der zum Zeitpunkt der Einführung des Patentierungsverbots herrschenden Unsicherheiten über die Patentierfähigkeit von Software, ausdrücklich eine Konkretisierung des Begriffs durch die Rechtsprechung und Lehre beabsichtigte.<sup>139</sup> Ein Computerprogramm stellt damit eine Erfindung im Sinne des § 1 I PatG dar, wenn es der Lösung eines konkreten technischen Problems dient, wobei unter Technik die planmäßige Benutzung oder Substitution beherrschbarer Naturkräfte außerhalb der menschlichen Verstandeskraft zur unmittelbaren Herbeiführung eines kausal übersehbaren Erfolges zu verstehen ist.

#### **(4) Vereinbarkeit mit Art. 27 TRIPS Übereinkommen**

Das am 01.01.1995 in Kraft getretene TRIPS<sup>140</sup> (Trade Related Aspects of Intellectual Property Rights) Übereinkommen ist das bis dato umfangreichste multilaterale Abkommen auf dem Gebiet des geistigen Eigentums. Es wurde im April 1994 in Marrakesch bei der sog. Uruguay Runde als Teil des Abschlussabkommens über die Gründung der WTO (World Trade Organisation) abgeschlossen. Das TRIPS Übereinkommen sieht die Einhaltung von gewissen

---

<sup>137</sup> *Horns*, GRUR 2001, 1 (8); *Kolle*, GRUR 1977, 58 (66); ebenso *Melullis*, FS-Erdmann, S. 414, allerdings primär auf § 1 II Nr. 1 PatG abstellend.

<sup>138</sup> Vgl. *Melullis*, FS-Erdmann, S. 414f.

<sup>139</sup> Vgl. BGH CR 2002, 88 (91) – „Suche fehlerhafter Zeichenketten“; *Schar*, Mitt. 1998, 322 (333).

<sup>140</sup> Abgedruckt in BGBl. II 1994, S. 1730ff.

Mindeststandards für sämtliche geistige Eigentumsrechte wie Urheberrechte, Patente oder Warenzeichen vor. Als multilaterales Abkommen sind die Bestimmungen für alle 146 Mitglieder (Stand: Juni 2003) der WTO verpflichtend. Da das TRIPS Übereinkommen nur einen Mindeststandard festlegt, sind die Mitgliedsstaaten gemäß Art. 1 I TRIPS nicht daran gehindert einen über die Bestimmungen des TRIPS Übereinkommens hinausgehenden Schutz festzulegen, sofern dieser den Regeln des TRIPS Übereinkommens nicht zuwiderläuft. Die Bestimmungen des TRIPS Übereinkommens sind daher auch bei Diskussion über den Patentschutz für Computerprogramme zu berücksichtigen. Gemäß Art. 27 I 1 des TRIPS Übereinkommens müssen Patente für Erfindungen auf allen Gebieten der Technik erhältlich sein, vorausgesetzt sie sind neu, beruhen auf einer erfinderischen Tätigkeit und sind gewerblich anwendbar. Art. 27 I 2 sieht ferner vor, dass bei der Frage des Patentschutzes nicht nach dem Gebiet der Technik diskriminiert werden darf. Teilweise wird das in § 1 II Nr. 3, III PatG bzw. Art. 52 II c), III EPÜ verankerte Patentierungsverbot für Computerprogramme „als solche“ als Verstoß gegen Art. 27 TRIPS Übereinkommen angesehen.<sup>141</sup> Ausgehend von der Prämisse, dass jedes Computerprogramm technischen Charakter habe, müsse in Einklang mit Art. 27 TRIPS Übereinkommen auch jedem Programm Patentschutz gewährt werden.<sup>142</sup> Eine Ausnahme für Computerprogramme „als solche“ sei daher unzulässig.<sup>143</sup> Diese Ansicht überzeugt jedoch nicht. Zwar verpflichtet Art. 27 TRIPS Übereinkommen die Mitgliedsstaaten zur Gewährung von Patenten für Erfindungen auf allen Gebieten der Technik. Die Frage was unter einer Erfindung und was unter dem Begriff der Technik zu verstehen ist, wird im TRIPS Übereinkommen jedoch nicht geregelt. Die Auslegung dieser Begriffe ist daher weiterhin Aufgabe des nationalen Gesetzgebers, wobei gemäß Art. 1 Nr. 1 TRIPS der Sinn und Zweck des Übereinkommens nicht konterkariert werden darf. Ziel des TRIPS Übereinkommens ist es, „einen angemessenen und wirksamen Schutz der Rechte des geistigen Eigentums zu fördern sowie sicherzustellen“.<sup>144</sup> In Bezug auf Patente soll sich dieser Schutz

---

<sup>141</sup> *Betten*, GRUR 1995, 775 (788); ders. GRUR 2000, 1009 (1010); *Schiurma*, GRUR Int. 1998, 852 (855); *Schmidtchen*, Mitt. 1999, 281 (282); zweifelnd *Ohly*, CR 2001, 809 (812).

<sup>142</sup> *Schiurma*, GRUR Int. 1998, 852 (855); *Schmidtchen*, Mitt. 1999, 281 (282).

<sup>143</sup> *Betten*, GRUR 1995, 775 (788).

<sup>144</sup> Präambel des TRIPS-Übereinkommens.

jedoch nur auf Erfindungen technischer Natur erstrecken. Sofern also unter Heranziehung des § 1 II Nr. 3, III PatG gefordert wird, dass ein Computerprogramm nur dann patentfähig ist, wenn es der Lösung eines konkreten technischen Problems dient, ist darin kein Verstoß gegen Art. 27 TRIPS Übereinkommen zu erblicken, sondern eine zulässige Auslegung der Begriffs der Technik, welche auch nicht mit dem Sinn und Zweck des TRIPS Übereinkommens in Widerspruch steht.<sup>145</sup> Auch der Gesetzgeber ging bei der Ratifizierung des TRIPS Übereinkommens davon aus, dass die Regelung des § 1 II Nr. 3, III PatG nicht im Widerspruch zu Art. 27 TRIPS steht.<sup>146</sup> Im Ergebnis stellt § 1 II Nr. 3, III PatG daher keinen Verstoß gegen Art. 27 TRIPS Übereinkommen dar.

#### **b. Neuheit, § 1 i.V.m. § 3 PatG**

Gemäß § 3 I PatG gilt eine Erfindung als neu, wenn sie nicht zum Stand der Technik gehört. Der Stand der Technik umfasst alles, was der Öffentlichkeit zu einem bestimmten Zeitpunkt zugänglich ist. An der Neuheit einer Erfindung fehlt es, wenn die beanspruchte Lehre irgendwann vor dem für den Zeitrang der Patentanmeldung maßgeblichen Tag irgendwo auf der Welt in irgendeiner Art und Weise der Öffentlichkeit zugänglich gemacht worden ist.<sup>147</sup> Hinsichtlich des Begriffs der Öffentlichkeit kann auf § 15 III UrhG zurückgegriffen werden. Danach ist Öffentlichkeit eine Mehrzahl von Personen, die weder bestimmt abgegrenzt noch persönlich untereinander verbunden sind. Die Veröffentlichung kann in jeder Weise erfolgen, sei es schriftlich oder mündlich, durch Vorbenutzung oder in sonstiger Weise. Es kommt dabei nicht darauf an, ob tatsächlich von einer Lehre Kenntnis genommen wurde. Es genügt vielmehr die bloße reale Möglichkeit zur Kenntnisnahme.<sup>148</sup> Bezogen auf Computerprogramme bedeutet dies, dass jedenfalls solche Software, die im Quellcode irgendwo zugänglich war (z.B. im Internet) zum Stand der Technik gehört und damit nicht mehr neu ist. Es fragt sich aber, ob dies auch für Programme gilt, die ausschließlich im Maschinencode vorliegen. Jedenfalls bei umfangreicheren Programmen lässt sich aus dem

---

<sup>145</sup> So auch BPatG, CR 2001, 155 (158) „Fehlerhafte Zeichenketten“; *König*, GRUR 2001, 577 (581); *Straus*, GRUR Int. 1996, 179 (191); *Tauchert*, GRUR 1999, 829 (831).

<sup>146</sup> Vgl. BT-Drucks. 12/7655, S. 345.

<sup>147</sup> *Benkhard-Ullmann*, PatG, § 3, Rn. 38; *Busse-Keukenschrijver*, PatG, § 3, Rn. 76.

<sup>148</sup> BGH GRUR 1971, 214 – „customer prints“; *Benkhard-Ullmann*, PatG, § 3, Rn. 53; *Schulte*, PatG, § 3, Rn. 51.

Binärcode eines Computerprogramms nicht ohne weiteres dessen Funktionsweise erschließen. Dies erfordert in der Regel eine langwierige und komplizierte Dekompilierung des Programms. Diese wiederum ist nach § 69e UrhG nur in engen Grenzen, nämlich zur Herstellung der Interoperabilität mit anderen Programmen, erlaubt. Daraus ließe sich folgern, dass Programme in kompilierter Form nicht zum Stand der Technik zu zählen sind, da die im Code verankerte Funktionalität der Öffentlichkeit nicht zugänglich ist.<sup>149</sup> Hiergegen sprechen m.E. jedoch zwei entscheidende Gesichtspunkte. Zum einen erfordert § 3 PatG nicht, dass tatsächlich Kenntnis von einer Lehre genommen wurde. Auf eine tatsächliche Dekompilierung kommt es daher gar nicht an. Entscheidender ist jedoch folgender Gesichtspunkt: Das Patentrecht schützt die einer Erfindung zugrunde liegende Idee und Funktionalität. Es kommt daher bei einem Computerprogramm im Gegensatz zum urheberrechtlichen Schutz gar nicht darauf an, wie das Programm im Einzelnen auf der Ebene des Codes aufgebaut ist. Der Schutz des Patentrechts erstreckt sich auf jedes Programm mit der gleichen Funktionalität wie das patentierte Programm, unabhängig davon, ob ein ähnlicher oder vollkommen anderer Code verwendet wurde. Dieser Umstand muss aber auch bei der Frage nach dem Stand der Technik berücksichtigt werden. Im Ergebnis zählen daher auch Programme, die nur in kompilierter Form zugänglich sind, zum Stand der Technik. Entscheidendes Problem bei der Ermittlung der Neuheit einer computerbezogenen Erfindung dürfte jedoch nach wie vor in den erheblichen Rechercheproblemen zu sehen sein.<sup>150</sup>

### **c. Erfinderische Tätigkeit, § 1 i.V.m. § 4 PatG**

Eine Erfindung gilt gemäß § 4 PatG als auf einer erfinderischen Tätigkeit beruhend, wenn sie sich für den Fachmann nicht in nahe liegender Weise aus dem Stand der Technik ergibt. Abzustellen ist auf das Können eines Durchschnittsfachmanns in dem Gebiet der beanspruchten Lehre.<sup>151</sup> Auf einer erfinderischen Tätigkeit beruht also nur das, was nicht von einem Durchschnittsfachmann zu erwarten war. Auf welchen Fachmann hierbei abzustellen ist, hängt von der beanspruchten Lehre ab. Handelt es sich beispielsweise um ein

---

<sup>149</sup> In diese Richtung argumentierend *Horns*, GRUR 2001, 1 (13).

<sup>150</sup> Siehe dazu unten V.3.a.aa.

<sup>151</sup> BGH GRUR 1953, 120 (122) – „Rohrschelle“; BGH GRUR 1954, 107 (110) – „Mehrfachschelle“; *Benkhard-Bruchhausen*, PatG, § 4, Rn. 10.

Betriebssystem, so ist auf den durchschnittlichen Softwareentwickler abzustellen. Handelt es sich hingegen um ein Programm zur Steuerung eines mechanischen Prozesses, so ist auf einen auf dem betreffenden Gebiet tätigen Techniker abzustellen. Problematisch ist die Frage, ob bei Erfindungen, die Merkmale technischer Natur mit Merkmalen nichttechnischer Natur verknüpfen, die erfinderische Tätigkeit auch in der nichttechnischen Leistung liegen kann. Der BGH hat dazu in seiner „*Tauchcomputer*“ Entscheidung festgestellt, dass der Erfindungsgegenstand nicht zerlegt und dann nur der Teil der Erfindung auf erfinderische Tätigkeit geprüft werden darf, der aus den technischen Merkmalen besteht.<sup>152</sup> In seiner späteren Entscheidung „*Sprachanalyseeinrichtung*“ hat der BGH dieses Ergebnis zwar wieder in Zweifel gezogen, die Frage im Ergebnis aber offen gelassen.<sup>153</sup> Das BPatG ist der Ansicht, dass die erfinderische Leistung auf dem Gebiet der Technik liegen muss, dass nichttechnische Elemente aber insoweit herangezogen werden können, als sie einen technischen Bezug aufweisen.<sup>154</sup> Begründet wird dies mit dem Schutzzweck des Patentrechts. Dem ist zuzustimmen. Sinn und Zweck des Patentrechts ist es, Problemlösungen auf dem Gebiet der Technik zu fördern. Diesem Ziel würde es aber widersprechen, wenn auch Erfindungen Patentschutz erlangen könnten, deren erfinderische Leistung ausschließlich im nichttechnischen Bereich angesiedelt ist. Die erfinderische Leistung muss daher auch bei computerbezogenen Erfindungen auf dem Gebiet der Technik liegen.<sup>155</sup> Auch hier stellt sich aber wiederum das Problem der mangelnden Recherchemöglichkeiten.<sup>156</sup>

#### **d. Gewerbliche Anwendbarkeit, § 1 i.V.m. § 5 PatG**

Eine Erfindung gilt nach § 5 I PatG als gewerblich anwendbar, wenn ihr Gegenstand auf irgendeinem gewerblichen Gebiet einschließlich der Landwirtschaft hergestellt oder benutzt werden kann. Ausschlaggebend ist nicht, ob eine gewerbliche Anwendung tatsächlich beabsichtigt ist, sondern nur ob die theoretische

<sup>152</sup> BGH GRUR 1992, 430 (432) – „*Tauchcomputer*“.

<sup>153</sup> BGH CR 2000, 500 (501f.) – „*Sprachanalyseeinrichtung*“.

<sup>154</sup> BPatG CR 2002, 559 (560) – „*Kein Patentanspruch für Cyber-Cash-Verfahren*“; vgl. auch Beschwerdekammer des EPA, GRUR Int. 2002, 87ff. – „*Steuerung eines Pensionssystems*“; Beschwerdekammer des EPA, Mitt. 2003, 123ff. – „*SIM-Card*“.

<sup>155</sup> So auch *Anders*, GRUR 2001, 555 (560); *Ohly*, CR 2001, 809 (816); *Sedlmeier*, Mitt. 2002, 55 (58); *Tauchert*, JurPC Web-Dok. 28/2002, Abs. 54; a.A. offenbar *Nack*, GRUR Int. 2000, 853 (856).

<sup>156</sup> Vgl. dazu unten V.3.a.aa.

Möglichkeit einer solchen besteht.<sup>157</sup> Für Computerprogramme wird in der Regel von einer gewerblichen Anwendbarkeit auszugehen sein, so dass sich hinsichtlich der Patentierbarkeit von Computerprogrammen diesbezüglich keine besonderen Probleme stellen.<sup>158</sup>

#### **e. Sonderfall: Computerbezogene Geschäftsmethoden**

Eng mit der Frage der Patentierbarkeit von Software verknüpft ist auch die Frage nach der Patentierbarkeit einer computerbezogenen Geschäftsmethode zu sehen. Nach den oben herausgebildeten Kriterien für die Patentierbarkeit von computerbezogenen Erfindungen, stellen sich hier jedoch keine besonderen Probleme. Das in § 1 II Nr. 3, III PatG enthaltene Patentierungsverbot für Pläne, Regeln und Verfahren für geschäftliche Tätigkeiten erklärt sich, wie bereits festgestellt wurde, daraus, dass eine Geschäftsmethode „als solche“, als eine Anweisung an den menschlichen Geist, keinen technischen Charakter aufweist. Eine Geschäftsmethode „als solche“ kann daher keinen Patentschutz genießen.<sup>159</sup> Zieht man nun die oben zur Patentierbarkeit eines Computerprogramms herausgearbeiteten Grundsätze heran, so ergibt sich daraus, dass sich auch bei einer computerbezogenen Geschäftsmethode, deren Technizität nicht allein daraus ergeben kann, dass sie mittels eines Computerprogramms auf einem Rechner ausgeführt wird.<sup>160</sup> Denn der technische Charakter eines Computerprogramms an sich reicht für eine Patentierbarkeit gerade nicht aus. Die Tatsache, dass eine Geschäftsmethode in Form eines Computerprogramms implementiert wird, ändert ihren Charakter als Anweisung an den menschlichen Geist nicht. Nur wenn die Geschäftsmethode (auch) der Lösung eines konkreten technischen Problems dient, ist sie dem Patentschutz zugänglich. Lässt sich die Technizität einer computerbezogenen Geschäftsmethode bejahen, so stellt sich in einem nächsten Schritt die Frage nach der Neuheit und der erfinderischen Tätigkeit. Die bloße Reproduktion einer bekannten Geschäftsmethode als Computerprogramm wird diese Anforderungen jedenfalls nicht erfüllen.<sup>161</sup>

---

<sup>157</sup> *Benkhard-Bruchhausen*, PatG, § 5, Rn. 3; *Schulte*, PatG, § 5, Rn. 3.

<sup>158</sup> *Kraßer* in: *Lehmann*, Rechtsschutz von Computerprogrammen, S. 224, Rn. 7.

<sup>159</sup> Vgl. dazu BGH GRUR 1958, 602ff. – „Wettschein“.

<sup>160</sup> BPatG CR 2002, 716 (717) – „Keine Patentierbarkeit Geschäftlicher Tätigkeit“; *Anders*, GRUR Int. 2001, 555 (560); *Melullis*, FS-Erdmann, S. 416.

<sup>161</sup> *Jänich*, GRUR 2003, 483 (487); *Anders*, GRUR Int. 2001, 555 (559); *Nack*, GRUR Int. 1998, 853 (858).

## **2. Aktuelle Entwicklungen auf Europäischer Ebene**

Als Mitglied der Europäischen Patentorganisation (EPO) und der EU sind bei der Untersuchung der Patentierbarkeit von computerbezogenen Erfindungen in Deutschland auch die jüngsten Entwicklungen auf europäischer Ebene von Bedeutung.

### **a. Revision des Europäischen Patentübereinkommens (EPÜ 2000)**

Am 29. November 2000 wurde auf der Diplomatischen Konferenz der Vertragsstaaten in München eine Revision (EPÜ 2000) des Europäischen Patentübereinkommen (EPÜ) beschlossen. Ziel war es, das nunmehr fast 30 Jahre alte Übereinkommen den technischen und rechtlichen Entwicklungen anzupassen.<sup>162</sup> Unter anderem wurde Art. 52 I EPÜ um die Worte „auf allen Gebieten der Technik“ ergänzt. Daraus haben sich jedoch keine Änderungen gegenüber der geltenden Rechtslage in Deutschland ergeben. Die geänderte Fassung des Art. 52 I EPÜ stellt nun lediglich ausdrücklich klar, was seit jeher als Grundsatz des Patentrechts angesehen wurde, nämlich dass nur technische Erfindungen Patentschutz genießen sollen. Außerdem wurde damit auch eine Angleichung an Art. 27 I TRIPS Übereinkommen erreicht. Eine weitere, wesentlich weitreichendere Änderung fand ihren Weg jedoch nicht in die endgültige Fassung des Vertragstexts. Entgegen der Forderung des sog. Basisvorschlags<sup>163</sup> wurden die Programme für Datenverarbeitungsanlagen nicht aus dem Negativkatalog des Art. 52 II c) gestrichen. Die Konferenz einigte sich darauf, dass angesichts des Stellenwerts dieser Frage die geltenden Bestimmungen des EPÜ in Bezug auf die Patentierbarkeit von Software vorerst unangetastet bleiben sollten, um so den weiteren Entwicklungen auf EU-Ebene nicht vorzugreifen.<sup>164</sup> Insgesamt ergeben sich aus der Revision des EPÜ damit keine Änderungen in Bezug auf die Patentierbarkeit von Software nach deutschem Recht.

---

<sup>162</sup> Der Text des EPÜ 2000 ist abrufbar unter [http://www.european-patent-office.org/epo/dipl\\_conf/pdf/dm00003a.pdf](http://www.european-patent-office.org/epo/dipl_conf/pdf/dm00003a.pdf).

<sup>163</sup> Der Text des sog. Basisvorschlags ist abrufbar unter [http://www.european-patent-office.org/epo/dipl\\_conf/pdf/dm00002a.pdf](http://www.european-patent-office.org/epo/dipl_conf/pdf/dm00002a.pdf).

<sup>164</sup> Entschließung der Konferenz; abrufbar unter [http://www.european-patent-office.org/epo/dipl\\_conf/pdf/dm00022.pdf](http://www.european-patent-office.org/epo/dipl_conf/pdf/dm00022.pdf).

## **b. Vorschlag für eine EU-Richtlinie zur Patentierbarkeit von computer-implementierten Erfindungen**

Am 20. Februar 2002 wurde nach fast 5-jähriger Vorbereitungszeit von der Europäischen Kommission ein Vorschlag für eine EU-Richtlinie zur Patentierbarkeit von computer-implementierten Erfindungen vorgelegt.<sup>165</sup> Nach der Veröffentlichung von Vorentwürfen seitens des Europäischen Rats<sup>166</sup> und des Europäischen Parlaments<sup>167</sup> hat der Ausschuss für Recht und Binnenmarkt (JURI) am 17. Juni 2003 einen geänderten Richtlinienvorschlag angenommen.<sup>168</sup> Eine endgültige Entscheidung des EU-Parlaments ist für September 2003 vorgesehen.

### **aa. Vorbereitende Schritte**

Die Vorbereitungen zur endgültigen Fassung des Vorschlags gehen zurück auf ein sog. Grünbuch aus dem Jahre 1997<sup>169</sup> und dessen Nachfolger aus dem Jahre 1999.<sup>170</sup> Im Zuge der Untersuchungen wurde von der Kommission festgestellt, dass die derzeitige Rechtslage aufgrund mangelnder Transparenz und fehlender Präzision zu einer undurchsichtigen und uneinheitlichen Rechtspraxis in Bezug auf die patentrechtliche Behandlung von computer-implementierten Erfindungen geführt habe.<sup>171</sup> Diese undurchsichtige und uneinheitliche Rechtspraxis wiederum schade dem gemeinsamen Binnenmarkt und führe zu erheblichen Wettbewerbsnachteilen in diesem Sektor.<sup>172</sup> Im Ergebnis wurde daher die Formulierung einer Richtlinie zur Harmonisierung der Rechtsvorschriften aller Mitgliedsstaaten beschlossen, um so eine einheitliche Anwendung und Auslegung der Rechtsvorschriften in allen Mitgliedsstaaten zu gewährleisten. Am 19. Oktober 2000 wurde von der Kommission eine umfangreiche

---

<sup>165</sup> Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates zur Patentierbarkeit von computer-implementierten Erfindungen, KOM (2002) 92; Text abrufbar unter: [http://europa.eu.int/comm/internal\\_market/en/indprop/comp/com02-92de.pdf](http://europa.eu.int/comm/internal_market/en/indprop/comp/com02-92de.pdf).

<sup>166</sup> Bericht des Ausschusses der Ständigen Vertreter für den Rat vom 08.11.2002; abrufbar unter <http://register.consilium.eu.int/pdf/de/02/st14/14017d2.pdf>.

<sup>167</sup> Vorentwurf des Ausschusses für Recht und Binnenmarkt vom 13.02.2003; abrufbar unter: <http://www.europarl.eu.int/meetdocs/committees/juri/20030219/488980.pdf>.

<sup>168</sup> Der Bericht des Ausschusses vom 17.06.2003 ist abrufbar unter: <http://swpat.ffii.org/news/03/juri0617/juri030617.pdf>.

<sup>169</sup> Grünbuch über das Gemeinschaftspatent und das Patentschutzsystem in Europa vom 24. Juni 1997, KOM (1997) 314; abrufbar unter: [http://europa.eu.int/comm/internal\\_market/en/indprop/patent/patde.pdf](http://europa.eu.int/comm/internal_market/en/indprop/patent/patde.pdf).

<sup>170</sup> Folgemaßnahmen zum Grünbuch über das Gemeinschaftspatent und das Patentschutzsystem in Europa vom 05. Februar 1999, KOM (1999) 42, abrufbar unter: [http://europa.eu.int/comm/internal\\_market/en/indprop/patent/8682de.pdf](http://europa.eu.int/comm/internal_market/en/indprop/patent/8682de.pdf).

<sup>171</sup> Folgemaßnahmen zum Grünbuch, S. 13.

<sup>172</sup> Folgemaßnahmen zum Grünbuch, S. 13ff.

Sondierung zur Frage der Patentierbarkeit von computerimplementierten Erfindungen eingeleitet und ein entsprechendes Konsultationspapier veröffentlicht.<sup>173</sup> Die betroffenen Kreise, die allgemeine Öffentlichkeit sowie die Mitgliedstaaten sollten auf Grundlage dieses Konsultationspapiers die Möglichkeit haben, zu der Frage der Patentierbarkeit von computerimplementierten Erfindungen Stellung zu nehmen. Bis zum Ende der Sondierung am 15. Dezember 2000 gingen insgesamt 1447 Antworten ein.<sup>174</sup> Die Auswertung der Antworten erfolgte durch das britische Meinungsforschungsinstitut PbT Consultants Ltd. und wurde in einem Abschlussbericht im Juli 2001 veröffentlicht.<sup>175</sup> Die Mehrzahl (91%) der Einsendungen war strikt gegen die Vorschläge der Kommission und generell für eine erhebliche Einschränkung des patentrechtlichen Schutzes für Computerprogramme. Die Gegner stammten zum Großteil aus dem Lager der Open Source Bewegung und die überwiegende Anzahl der Antworten wurde als Teil einer Kampagne der EuroLinux Allianz<sup>176</sup> übermittelt. Es wurde befürchtet, dass eine Ausweitung des Patentschutzes für Software, Innovationen und Wettbewerb auf dem Gebiet der Softwareentwicklung behindern und KMUs und Open Source Software-Entwickler unangemessen benachteiligen würde. Die Befürworter von Softwarepatenten stammten größtenteils aus dem Lager der Anwälte, großer Unternehmen und staatlicher Einrichtungen. Hier wurde überwiegend darauf abgestellt, dass Softwarepatente für den internationalen Wettbewerb und den Investitionsschutz unabdingbar seien.

#### **bb. Aufbau und Regelungsinhalt<sup>177</sup>**

Der Anwendungsbereich der Richtlinie<sup>178</sup> wird in Art. 1 RL abgesteckt. Danach befasst sich die Richtlinie mit der Patentierbarkeit von computerimplementierten Erfindungen. Darunter ist gemäß Art. 2 (a) RL jede Erfindung zu verstehen, zu deren Ausführung ein

---

<sup>173</sup> Die Patentierbarkeit computerimplementierter Erfindungen: Sondierungspapier der Dienststellen der Generaldirektion Binnenmarkt; abrufbar unter: [http://europa.eu.int/comm/internal\\_market/en/indprop/comp/softde.pdf](http://europa.eu.int/comm/internal_market/en/indprop/comp/softde.pdf).

<sup>174</sup> Die Antworten sind abrufbar unter: [http://europa.eu.int/comm/internal\\_market/de/indprop/comp/softreplies.htm](http://europa.eu.int/comm/internal_market/de/indprop/comp/softreplies.htm).

<sup>175</sup> Englische Fassung des Berichts abrufbar unter: [http://europa.eu.int/comm/internal\\_market/de/indprop/comp/softpatanalyse.htm](http://europa.eu.int/comm/internal_market/de/indprop/comp/softpatanalyse.htm).

<sup>176</sup> <http://www.eurolinux.org>.

<sup>177</sup> Vgl. dazu auch *Howard*, CRI 2002, 97ff.; *Metzger*, CR 2003, 313ff.; *Röttinger*, CR 2002, 616ff.; *Sedlmaier*, Mitt. 2002, 97ff.

<sup>178</sup> In der geänderten Fassung vom 17.06.2003.

Computer, ein Computernetz oder eine sonstige programmierbare Vorrichtung eingesetzt wird und die mindestens ein Merkmal aufweist, das ganz oder teilweise mit einem oder mehreren Computerprogrammen realisiert wird.

### **(1) Technizität einer computer-implementierten Erfindung**

Dass nur technische Erfindungen patentierbar sind, wird in der Richtlinie nicht ausdrücklich erwähnt. Aus Erwägungsgrund 10 ergibt sich aber, dass nur solche Erfindungen, die technischen Charakter haben und somit einem Gebiet der Technik zuzuordnen sind, als patentierbar gelten. Damit hält auch die Richtlinie an dem Erfordernis der Technizität fest. Folglich müssen nach der Richtlinie auch computer-implementierte Erfindungen technischer Natur sein, um patentierbar zu sein. Die äußerst umstrittene, ehemals in Art. 3 RL a.F. enthaltene Regelung, wonach jede computer-implementierte Erfindung per se als dem Gebiet der Technik zugehörig angesehen wird, wurde ersatzlos gestrichen. Die Streichung wurde damit begründet, dass der Artikel unnötig und seinem Geltungsbereich nach unklar sei. Außerdem könnte er herangezogen werden, um den Anwendungsbereich des Patentschutzes auszuweiten.<sup>179</sup> Wie sich aus dem geänderten Erwägungsgrund 12 der Richtlinie ergibt, geht der Ausschuss jedoch nach wie vor davon aus, dass jede computer-implementierte Erfindung einem Gebiet der Technik zuzurechnen ist. Nach der Begründung zur Richtlinie überwindet somit jede computer-implementierte Erfindung „die erste Hürde auf dem Weg zur Patentierbarkeit“.<sup>180</sup> Wie sich aus den folgenden Voraussetzungen ergibt, wird damit der Schwerpunkt der Prüfung vom Merkmal der Technizität auf das Merkmal der erfinderischen Tätigkeit verlagert.

### **(2) Voraussetzungen der Patentierbarkeit**

Nach Art. 4 Nr. 1 RL ist eine computer-implementierte Erfindung patentierbar, sofern sie gewerblich anwendbar und neu ist und auf einer erfinderischen Tätigkeit beruht. Damit ist Art. 4 Nr. 1 RL identisch mit der geltenden Rechtslage nach Art. 52 I EPÜ bzw. § 1 I PatG. Gemäß Art. 4 Nr. 2 RL ist die Voraussetzung der erfinderischen Tätigkeit nur dann erfüllt, wenn die computer-implementierte Erfindung einen technischen Beitrag leistet. Ein technischer Beitrag

---

<sup>179</sup> Vgl. Bericht des Ausschusses vom 17.06.2003, S. 14.

<sup>180</sup> Begründung der RL, S. 7.

wiederum wird in Art. 2 (b) RL definiert als ein Beitrag zum Stand der Technik auf einem Gebiet der Technik, der für eine fachkundige Person nicht nahe liegend ist. Damit entspricht die Regelung der Richtlinie den bereits oben dargestellten Auffassungen des BPatG und der Beschwerdekammern des EPA, welche ebenfalls fordern, dass die erfinderische Leistung auf einem Gebiet der Technik liegen muss.<sup>181</sup> Art. 4 Nr. 3 RL übernimmt die sog. Gesamtbetrachtungslehre, welche schon seit längerem vom BGH und vom EPA zur Ermittlung der erfinderischen Tätigkeit angewendet wird. Erwägungsgrund 13a und Art. 4a RL stellen nunmehr klar, dass ein technischer Beitrag nicht schon darin gesehen werden kann, dass zur Ausführung der Erfindung ein Computer zum Einsatz kommt. Gemäß Art. 4a RL sind „Erfindungen, zu deren Ausführung ein Computerprogramm eingesetzt wird und durch die Geschäftsmethoden, mathematische oder andere Methoden angewendet werden, nicht patentfähig, wenn sie über die normalen physikalischen Interaktionen zwischen einem Programm und dem Computer [...] keine technischen Wirkungen erzeugen.“ Damit entspricht die Regelung im Ergebnis der neueren Rechtsprechung des BGH, welcher die Lösung eines „konkreten technischen Problems“ fordert und des EPA, welches auf das Vorhandensein eines „weiteren technischen Effekt“ abstellt.<sup>182</sup> Allerdings wird die Prüfung dieser Voraussetzungen vom Merkmal der Technizität auf das Merkmal der erfinderischen Tätigkeit verlagert.

### **(3) Form des Patentanspruchs**

Art. 5 Nr. 1 RL sieht zwei Möglichkeiten für die Formulierung von Patentansprüchen für computer-implementierte Erfindungen vor. Eine computer-implementierte Erfindung kann entweder als Erzeugnisanspruch auf den programmierten Computer oder als Verfahrensanspruch auf das von einem Computer durch Ausführung der Software verwirklichte Verfahren angemeldet werden. Im Gegensatz zum ursprünglichen Vorschlag der Kommission gestattet Art. 5 Nr. 2 RL nun auch die Anmeldung eines Computerprogramms auf einem Datenträger oder als Signal, sofern das Programm, sobald es auf einem Computer geladen oder ausgeführt wird, einen Erzeugnis- oder Verfahrensanspruch im Sinne von Art. 5 Nr. 1 RL

---

<sup>181</sup> Siehe oben III.1.c.

<sup>182</sup> Vgl. BGH CR 2002, 88ff. – „Suche fehlerhafter Zeichenketten“; Beschwerdekammer des EPA, GRUR Int. 1999, 1053ff. – „Computerprogrammprodukt/IBM“.

darstellt und die Voraussetzungen der Art. 4 und 4a der Richtlinie erfüllt. Damit geht die Richtlinie konform mit der Rechtsprechung des BGH und des EPA zur Anmeldung eines sog. Computerprogrammprodukts.<sup>183</sup>

#### **(4) Regelungen zur Dekompilierung und Interoperabilität**

Art. 6 der Richtlinie stellt klar, dass die nach Art. 6 I der EG-Software-Richtlinie<sup>184</sup> (§ 69e UrhG) zur Dekompilierung erlaubten Maßnahmen durch die Vorschriften über die Patentierbarkeit computer-implementierter Erfindungen unberührt bleiben. Der neu vorgeschlagene Art. 6a stellt zudem klar, dass mit Patenten für computer-implementierte Erfindungen nicht die Interoperabilität zwischen Computersystemen oder Netzwerken verhindert werden darf.

#### **cc. Ergebnis**

Ausweislich der Begründung<sup>185</sup> und den Erwägungsgründen der Richtlinie, soll diese nicht zu einer Änderung der bestehenden Rechtslage bezüglich computer-implementierter Erfindungen in Europa führen. Betrachtet man die in den Art. 4 und 4a RL zugrunde gelegten Voraussetzungen der Patentierbarkeit, so spricht in der Tat wenig dafür, dass die Richtlinie zu einer Ausweitung des Patentschutzes für computer-implementierte Erfindungen führen könnte. Ihren Voraussetzungen nach entspricht die Richtlinie der derzeitigen Rechtslage. Zwar wird der Schwerpunkt der Prüfung vom Merkmal der Technizität auf das Merkmal der erfinderischen Tätigkeit verlagert, im Ergebnis entspricht der geforderte technische Beitrag jedoch der geltenden Interpretation von Art. 52 II c), III EPÜ bzw. § 1 II Nr. 3, III PatG. Die neu eingefügten Erwägungsgründe 13a – 13d und Erwägungsgrund 14 stellen ausdrücklich klar, dass Algorithmen an sich nicht patentierbar sind und auch eine Kombination mit technischen Aspekten nicht dazu führen kann, dass nicht-technische Inhalte patentierbar werden. In Bezug auf die Patentierbarkeit von computerbezogenen Geschäftsmethoden stellt die Richtlinie sogar ausdrücklich klar, dass diese grundsätzlich nicht

---

<sup>183</sup> BGH CR 2002, 88ff. – „Suche fehlerhafter Zeichenketten“; Beschwerdekammer des EPA, GRUR Int. 1999, 1053ff. – „Computerprogrammprodukt/IBM“.

<sup>184</sup> Richtlinie 91/250/EWG des Rates vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen, Abl. EG Nr. L 122, S. 42.

<sup>185</sup> Begründung der RL, S. 12.

patentierbar sind, sofern nicht auch ein technischer Beitrag geleistet wird. Im Ergebnis ist daher nach dem aktuellen Stand der Dinge nach einer Umsetzung der Richtlinie in nationales Recht keine Änderung der Rechtslage in Deutschland zur Patentierbarkeit von computerimplementierten Erfindungen zu erwarten.<sup>186</sup>

#### **IV. Das Open Source Software Modell**

Das Open Source Software Modell ist mehr als nur ein Modell zur Softwareentwicklung. Es ist eine Weltanschauung. Als solche unterscheidet es sich grundlegend von dem auf Geheimhaltung und den Schutz des geistigen Eigentums bedachte Modell der proprietären Softwareentwicklung. Hintergrund und Reichweite des besonderen Spannungsverhältnisses zwischen Open Source Software und der Patentierbarkeit von Software erschließen erst bei genauerer Betrachtung der entwicklungstechnischen und lizenzrechtlichen Besonderheiten des Open Source Software Modells. Unerlässlich ist auch ein Blick auf die Philosophie und die Entstehungsgeschichte der Open Source Bewegung. In diesem Abschnitt sollen daher die Besonderheiten des Open Source Software Modells im Vergleich zum proprietären Softwareentwicklungsmodell und zu anderen autodistributiven Entwicklungsmodellen dargestellt werden.

##### **1. Die Entstehungsgeschichte der Open Source Bewegung<sup>187</sup>**

Die Entstehung der Open Source Bewegung steht in direktem Zusammenhang mit der gewandelten Bedeutung von Software zu einem eigenständigen Wirtschaftsgut und dem damit einhergehenden Untergang der Hackerkultur.

##### **a. Der Bedeutungswandel von Software zu einem eigenständigen Wirtschaftsgut und der Niedergang der Hackerkultur**

In den Anfangstagen der Computerindustrie wurde Software lediglich als eine notwendige Beigabe zu Hardware und nicht als eigenständiges Wirtschaftsgut angesehen. Dementsprechend gab es auch nahezu keine kommerzielle Vermarktung von Software. Die „Big

---

<sup>186</sup> A.A. Metzger, CR 2003, 313 (317); Sedlmaier, Mitt. 2002, 97 (101).

<sup>187</sup> Umfassende Darstellungen finden sich bei Grassmuck, Freie Software, S. 202ff., Jaeger/Metzger, OSS, S. 7ff.; Schiffner, OSS, S. 57ff.

Player“ der damaligen Zeit wie IBM (System/360<sup>188</sup>) oder DEC (PDP-Serie<sup>189</sup>) verkauften ihren Kunden ein Gesamtpaket an Hardware, Dokumentation und Support und lieferten ihre raumfüllenden Monsterrechner direkt mit den entsprechenden Betriebssystemen aus. Programme wurden in der Regel mitsamt ihrem Quellcode ausgeliefert, um es jedem Nutzer zu ermöglichen, die Software an seine speziellen Bedürfnisse anzupassen und zu verbessern, oder eigene Programme zu entwickeln. Der freie und transparente Umgang mit dem Quellcode führte zu einem regen Informationsaustausch zwischen Nutzern und Hardwareherstellern, sowie zwischen den Nutzern untereinander. Insbesondere an den amerikanischen Universitäten und staatlichen Forschungseinrichtungen (insbesondere am MIT) entstand dadurch eine Gemeinschaft von sog. Hackern, welche sich der Optimierung und Weiterentwicklung der Software verschrieben hatte.<sup>190</sup>

Zu einer ersten Wende kam es jedoch Ende 1969, als IBM unvermittelt die Bündelung von Hard- und Software einstellte. Damit wurde das Fundament für die Entstehung einer eigenständigen Softwareindustrie gelegt. Es entstanden erste Unternehmen, welche sich ausschließlich mit der Programmierung von Anwendungsprogrammen beschäftigten. Je deutlicher sich Software zu einem eigenständigen Wirtschaftsgut entwickelte, desto unfreier wurde auch der Umgang mit selbiger. Die Konsequenzen für den universitären Forschungsrahmen waren verheerend. Wo früher ein freier Austausch von Ideen herrschte, wurden jetzt Forschende und Lehrende gezwungen, Kooperationen zu beschränken oder ganz zu unterlassen. Ende der 70er Jahre wurde Software zunehmend proprietär vermarktet, d.h. der Quellcode wurde geheim gehalten und der Nutzer musste eine Lizenz erwerben um das Programm benutzen zu können. Eine Veränderung oder gar eine Weiterverbreitung der Software wurde nicht mehr gestattet. Ein gutes Beispiel um den neuen Zeitgeist zu verdeutlichen bietet ein am 3. Februar 1976

---

<sup>188</sup> Vgl. dazu <http://www.wikipedia.org/wiki/System/360>.

<sup>189</sup> Vgl. dazu [http://www.wikipedia.org/wiki/Digital\\_Equipment\\_Corporation](http://www.wikipedia.org/wiki/Digital_Equipment_Corporation).

<sup>190</sup> Der Begriff des Hackers war damals keineswegs so negativ belegt wie heutzutage, sondern bezeichnete einen Menschen, der sich für die Funktionsweise eines Rechnersystems interessierte und Spaß daran hatte, dessen Grenzen auszudehnen. Vgl. dazu *Raymond*, *The New Hacker's Dictionary*; <http://www.hack.gr/jargon/>.

veröffentlichter offener Brief<sup>191</sup> von *Bill Gates*, der ein Jahr zuvor zusammen mit *Paul Allen* die Firma Microsoft gegründet hatte. In seinem Brief bezichtigte *Bill Gates* Nutzer, welche Software kopierten und tauschten des Diebstahls, und warf ihnen vor, dadurch die Entwicklung guter Software zu verhindern. Der geänderte Umgang mit Software zog legislatorische Schritte in den USA nach sich. 1974 berief der US-Kongress eine Kommission ein, die sich mit dem rechtlichen Status von Computerprogrammen befassen sollte. In ihrem Abschlussbericht empfahl die CONTU (Commission on New Technological Uses of Copyrighted Works) Computerprogramme als Sprachwerke unter den Schutz des Copyright zu stellen. 1980 wurde der Schutz von Software durch Copyright schließlich ausdrücklich im amerikanischen Recht verankert.<sup>192</sup> Im Jahre 1981 wurde dann in einem Urteil des US Supreme Court erstmals die Patentierbarkeit von Software bejaht.<sup>193</sup> Der endgültige Durchbruch der Softwareindustrie wurde wiederum durch eine folgenreiche Entscheidung von IBM eingeleitet. Mit der Einführung des Personal Computers (PC) am 12. August 1981 machte IBM den Computer und damit auch die dazugehörige Softwareindustrie dem Massenmarkt zugänglich.<sup>194</sup> Durch die Offenlegung der Architektur des PCs wurden bald unzählige IBM-PC Clones auf den Markt geworfen. Mit der Eröffnung des Massenmarktes für die Softwareindustrie kam es auch zum Untergang der Hackerkultur wie sie bis dahin herrschte. Immer mehr ehemalige Hacker wurden von Softwarefirmen angeworben oder gründeten eigene Softwarefirmen. Die Ära der Hackerkultur neigte sich endgültig ihrem Ende entgegen.

## **b. Das UNIX Betriebssystem**

Die Entstehungsgeschichte der Open Source Bewegung ist eng verknüpft mit der Geschichte des UNIX Betriebssystems. UNIX entstand aus dem Multics (MULTiplexed Information and Computing Service) Betriebssystem, an dem Programmierer des MIT, General Electric und AT&T seit 1964 gearbeitet hatten. Aufgrund der Komplexität und Fehleranfälligkeit von Multics ließen brauchbare

---

<sup>191</sup> *Bill Gates*, "An open Letter to Hobbyists"; <http://www.blinkenlights.com/classiccmp/gateswhine.html>.

<sup>192</sup> Vgl. Title 17, Sektion 101 und 117 United States Code (USC); abrufbar unter: <http://www.copyright.gov/title17/>.

<sup>193</sup> *Diamond v. Diehr* 450 U.S. 175, GRUR 1981, 646ff.

<sup>194</sup> *Grassmuck*, Freie Software, S. 204; *Jaeger/Metzger*, OSS, S. 9.

Ergebnisse aber auf sich warten. 1969 zog sich AT&T aus dem gemeinsamen Projekt zurück.<sup>195</sup> *Ken Thompson*, ein Mitarbeiter am Multics Projekt, startete jedoch noch im selben Jahr an den AT&T Bell Laboratories einen neuen Versuch, ein Time-Sharing<sup>196</sup> fähiges Betriebssystem zu entwickeln. Bei der Entwicklung von UNIX<sup>197</sup> wurde aus den Fehlern von Multics gelernt. Anstatt aus einem einzigen allumfassenden Programm zu bestehen, wurde UNIX derart konzipiert, dass es aus einer Vielzahl von kleinen, unabhängigen Werkzeugen (sog. Modulen) aufgebaut wurde. Dieser modulare Aufbau sollte sich später für die Entwicklung von Open Source Software als entscheidender Vorteil herausstellen. Um eine Portierbarkeit auf andere Computersysteme zu ermöglichen, wurde UNIX 1971 in der eigens dazu entwickelten Programmiersprache C vollständig neu geschrieben. Noch im selben Jahr wurde UNIX offiziell zum internen Standardbetriebssystem von AT&T. Aus kartellrechtlichen Gründen war es dem Telefonmonopolisten AT&T jedoch untersagt, UNIX kommerziell zu vertreiben. Stattdessen lizenzierte die Firma das Betriebssystem zu einem nominellen Betrag an Universitäten und andere staatliche Einrichtungen. Dort wurde UNIX aufgrund seiner Modularität und leichten Zugänglichkeit von interessierten Studenten und Informatikern schnell weiterentwickelt und verbessert. Insbesondere die *Berkeley University of California* übernahm eine Vorreiterrolle und veröffentlichte 1977 mit BSD (Berkeley Software Distribution) erstmals eine eigene UNIX-Distribution. Als AT&T 1984 in einem Kartellrechtsverfahren in mehrere konkurrierende Unternehmen zerschlagen wurde, konnte das eigens dafür gegründete Tochterunternehmen UNIX System Laboratories (USL) erstmals in großem Stile mit einer kommerziellen Vermarktung von UNIX beginnen. Die daraufhin geänderten Lizenzbedingungen verwandelten das ehemals freie UNIX zu einem proprietären Softwareprogramm, welches nur noch gegen erhebliche Lizenzgebühren zu erhalten war.<sup>198</sup>

---

<sup>195</sup> Multics wurde ständig weiterentwickelt und sogar bis zum Jahre 2000 eingesetzt. Vgl. zur weiteren Entwicklung von Multics <http://www.multicians.org/history.html>.

<sup>196</sup> Ein Time-Sharing fähiges Betriebssystem ermöglicht die gleichzeitige und unabhängige Nutzung eines Computers durch mehrere Nutzer. Time-Sharing ist der Vorläufer des heutigen Begriffs Multi-Tasking.

<sup>197</sup> Ursprünglich Unics buchstabiert, ein Akronym für „UNiplexed Information and Computing Service“.

<sup>198</sup> Vgl. zur weiteren Entwicklung von UNIX, *Grassmuck*, Freie Software, S. 211ff.

### c. Das GNU/Linux-Projekt

Es war zu dieser Zeit, also Anfang 1984 als *Richard Stallmann*, ein Mitarbeiter des MIT, mit der Entwicklung eines eigenen UNIX-kompatiblen Betriebssystems begann.<sup>199</sup> Als Hacker der alten Schule störte sich *Stallmann* an der zunehmenden Geheimhaltungspraxis und den Reglementierungsbestrebungen der noch jungen Softwareindustrie im Umgang mit dem Quellcode ihrer Programme. Sein Bestreben war es ein alternatives Softwaremodell zu entwickeln, dass jedem einen freien Zugang und eine ungehinderte Weiterentwicklung der Software gestattete. Grundlage dieses Modells sollte ein UNIX-kompatibles, aber von lizenzpflichtigen Code völlig bereinigtes Betriebssystem sein. *Stallmann* nannte das Projekt GNU, ein aus dem rekursiven Akronym „GNU's not Unix“ zusammengesetzter Name. Um seine Vision von einem freien Softwaremodell zu verwirklichen gründete *Stallmann* 1985 die gemeinnützige Free Software Foundation (FSF).<sup>200</sup> Aufgabe der FSF war es die politische, organisatorische und finanzielle Basis für das GNU-Projekt zu liefern. Zusammen mit *Eben Moglen*, einem Jura Professor der Columbia University, entwickelte *Stallmann* 1989 die GNU General Public License (GPL), die grundlegende Lizenz für die Nutzung von sämtlichen im Rahmen des GNU-Projekts entwickelten Programme. Dank einer immer größer werdenden Gemeinde von Programmierern und Unterstützern wuchs das GNU-Projekt schnell heran. Vor allem der modulare Aufbau von UNIX gestattete es, einzelne Bestandteile des neuen Betriebssystems unabhängig voneinander zu entwickeln und so die Arbeit unter vielen autonom handelnden Programmierern aufzuteilen.<sup>201</sup> Anfang der 90er Jahre war das GNU Betriebssystem weitgehend fertig gestellt. Es fehlte jedoch noch eine entscheidende Komponente um ein vollwertiges Betriebssystem zu erhalten, der sog. Kernel.<sup>202,203</sup>

---

<sup>199</sup> Vgl. den in einer Newsgroup verschickten Aufruf *Stallmann's* zur Mitarbeit an dem Projekt <http://www.fsf.org/gnu/initial-announcement.html>.

<sup>200</sup> <http://www.fsf.org>.

<sup>201</sup> Häufig bestand der einzige Kontakt zwischen Mitarbeitern des Projekts in dem Austausch von E-Mails.

<sup>202</sup> Der Kernel ist der Kern eines Betriebssystems, welcher die elementare Prozess- und Datenorganisation festlegt, auf die alle weiteren Softwarebestandteile zurückgreifen. Vgl. [http://www.wikipedia.org/wiki/Kernel\\_\(computers\)](http://www.wikipedia.org/wiki/Kernel_(computers)).

<sup>203</sup> Der vom GNU-Projekt ab 1990 entwickelte Kernel HURD stellte sich als ein so ehrgeiziges Projekt heraus, dass selbst heute noch keine finale Version (aktuell 0.2) verfügbar ist. Vgl. zu HURD <http://www.gnu.org/software/hurd/hurd.html>.

Der Zufall wollte es, dass just zu dieser Zeit (August 1991) der finnische Informatikstudent *Linus Torvalds* mit der Programmierung eines eigenen UNIX-kompatiblen Betriebssystems für seinen 386er PC begonnen hatte.<sup>204</sup> Am 5. Oktober 1991 stellte *Torvalds* mit Version 0.02 eine erste lauffähige Version seines Linux (Linus' UNIX) Kernels mitsamt Quellcode anderen interessierten Programmieren im Internet zur Verfügung und bat um Mithilfe bei der Fehlerbeseitigung und Verbesserung des Programms. Dank der Mithilfe und den Verbesserungsvorschlägen der Internetgemeinde und dem Vorgehen *Torvalds'*, der diese rasch umsetzte und jede neue Version sofort wieder ins Netz stellte, entwickelte sich Linux schnell zu einem leistungsstarken und stabilen Kernel. Version 0.12 stellte *Torvalds* dann im Januar 1992 unter die GNU GPL. Durch Rückgriff auf die bereits weitgehend fertig gestellten Programme und Werkzeuge des GNU-Projekts und andere freie Softwareprojekte (z.B. XFree86) konnte schnell ein professionelles Betriebssystem mit enormen Funktionsumfang entstehen. Im März 1994 wurde schließlich die erste stabile Version 1.0 des GNU/Linux Betriebssystems freigegeben.<sup>205</sup> Auch wenn sich der Name Linux als Synonym für das gesamte Betriebssystem durchgesetzt hat, sollte angesichts der Tatsache, dass die Komponenten des fertigen Betriebssystems größtenteils aus dem GNU-Projekt stammen richtigerweise vom GNU/Linux Betriebssystem gesprochen werden.<sup>206</sup>

#### **d. Die wirtschaftliche Bedeutung von Open Source Software**

War die Open Source Bewegung bis Mitte der 90er Jahre lediglich einem eingeschworenen Kreis von Idealisten und Hobby-Programmierern zugänglich, so änderte sich die öffentliche Wahrnehmung dieses Phänomens schlagartig mit dem Auftreten des GNU/Linux Betriebssystems. Insbesondere der ab 1993 einsetzende kommerzielle Vertrieb von GNU/Linux Distributionen durch Unternehmen wie Debian, SuSE, Red Hat oder Mandrake, machte das System auch einer breiten Masse von Privatnutzern zugänglich. So erhält der Erwerber einer solchen Distribution ein umfangreiches Softwarepaket bestehend aus Betriebssystem, Anwender-

---

<sup>204</sup> Erste Beiträge dazu tauchten im Juli 1991 in der Usenet Newsgroup comp.os.minix auf, vgl. <http://www.li.org/linuxhistory.php>.

<sup>205</sup> Die letzte stabile Version des Linux-Kernels ist Version 2.4.21 vom 13.06.2003; siehe <http://www.kernel.org>.

<sup>206</sup> *Jaeger/Metzger*, OSS, S. 13; vgl auch <http://www.fsf.org/gnu/linux-and-gnu.html>.

programmen, diversen Tools und Treibern, einer bedienerfreundlichen Installationsroutine, einer umfangreichen Dokumentation und einem umfassenden Support. Daneben führte vor allem die Entwicklung der mit Windows 9x vergleichbaren grafischen Benutzeroberflächen KDE (K-Desktop Enviroment) und GNOME (GNU Network Objekt Model Enviroment) dazu, Berührungängste mit dem GNU/Linux Betriebssystem abzubauen. Heute gibt es schätzungsweise 30 Millionen GNU/Linux-Installationen weltweit.<sup>207</sup> Zwar hält GNU/Linux im Desktop-Markt mit gerade mal 2% einen noch verschwindend geringen Marktanteil (gegenüber 92% Marktanteil von Microsoft), bei den Serverbetriebssystemen liegt GNU/Linux mit mittlerweile 26% Marktanteil aber schon an zweiter Stelle hinter Windows NT/2000 mit 44% Marktanteil.<sup>208</sup> Große Unternehmen wie Edeka, Sixt, Debis und Ikea haben ihre EDV-Infrastruktur bereits vollständig auf GNU/Linux umgestellt.<sup>209</sup> Auch in der öffentlichen Verwaltung wird zunehmend mit GNU/Linux gearbeitet.<sup>210</sup> So wurde vom Bundestag am 15.02.2001 einem Antrag der Regierungsparteien zugestimmt, künftig Open Source Software besonders zu fördern und in die Bundesverwaltung einzuführen.<sup>211</sup> Mittlerweile gibt es eine unübersehbare Fülle an Open Source Software für nahezu jeden Bereich und jedes Betriebssystem.<sup>212</sup> Insbesondere die Infrastruktur des Internets besteht zum größten Teil aus Open Source Software. Von immenser Bedeutung ist die DNS-Server Software Bind, welche bei 95% aller DNS-Server eingesetzt wird. Nach der jüngsten Erhebung von *Netcraft Ltd.* im Mai 2003 werden fast 63% aller Web Server mit der Open Source Software Apache betrieben.<sup>213</sup> Mit der Open Source Software Sendmail werden nahezu 40% des gesamten E-Mail Verkehrs abgewickelt.<sup>214</sup> Weitere wichtige Open Source

---

<sup>207</sup> *Grassmuck*, Freie Software, S. 229.

<sup>208</sup> So eine Studie von IDC, vgl. <http://www.heise.de/newsticker/data/ola-15.05.03-002>.

<sup>209</sup> *Metzger/Jaeger*, GRUR Int. 1999, 839 (840).

<sup>210</sup> So wurde jüngst vom *Münchener* Stadtrat entschieden alle 14.000 Rechner der Stadt künftig auf GNU/Linux umzustellen, vgl. <http://www.heise.de/newsticker/data/anw-28.05.03-004/>. Auch die Stadt *Schwäbisch-Hall* will bis 2004 ihr EDV-Infrastruktur komplett auf GNU/Linux umstellen, vgl. <http://www.heise.de/newsticker/data/anw-26.02.03-006/>.

<sup>211</sup> BT-Drucks. 14/5246, S. 4f.

<sup>212</sup> Eine Übersicht über aktuelle Open Source Programme findet sich z.B. auf <http://freshmeat.net>.

<sup>213</sup> Vgl. <http://news.netcraft.com>.

<sup>214</sup> Vgl. <http://liquidzero.net/surveys/smtp/latest/>. Die von *Grassmuck*, Freie Software, S. 260 genannte Zahl von 90% erscheint etwas zu hoch gegriffen.

Projekte sind die Datenbanksoftware MySQL<sup>215</sup>, das Office-Paket OpenOffice<sup>216</sup>, das Grafikbearbeitungsprogramm GIMP (GNU Image Manipulation Program)<sup>217</sup> und die Skriptsprache PERL<sup>218</sup>. Neben der Softwareentwicklung hat sich insbesondere ein großer Markt auf dem Dienstleistungssektor entwickelt.<sup>219</sup> Die Bedeutung von Open Source Software wurde mittlerweile auch von großen Computerherstellern erkannt. So liefern inzwischen Unternehmen wie IBM, Hewlett-Packard oder auch der PC-Discounter Vobis ihre Produkte mit vorinstalliertem GNU/Linux aus. Auch der Bereich der sog. Embedded Systeme<sup>220</sup> stellt einen großen Wachstumsmarkt für Open Source Software dar.

## 2. Zum Begriff der Open Source Software

Die wörtliche Übersetzung von Open Source Software bedeutet „quelloffene Software“, allerdings stellt die Offenbarung des Quellcodes nur ein - wenn auch notwendiges - Charakteristikum von Open Source Software dar. Des weiteren zeichnet sich das Open Source Software Modell dadurch aus, dass dem Nutzer eine größtmögliche Freiheit im Umgang mit einer Software eingeräumt wird. So erhält jeder Nutzer eines Programms das unentgeltliche Recht, die Software zu vervielfältigen, zu bearbeiten und zu verbreiten. Darüber hinaus wird jeder Nutzer angeregt, eventuell vorgenommene Veränderungen der Gemeinschaft wiederum zur Verfügung zu stellen, damit auch andere von den Verbesserungen profitieren können.<sup>221</sup> Wie in den Anfangstagen der Softwareentwicklung basiert das Open Source Entwicklungsmodell also auf einem frühzeitigen und uneingeschränkten Wissenstransfer. Gerade in diesem Wissenstransfer liegt die unbestrittene Stärke des Open Source Software Modells. Aufgrund der dadurch ermöglichten Synergieeffekte weisen Open Source Programme ein Maß ab Stabilität und Sicherheit auf, welches das von herkömmlicher proprietärer

---

<sup>215</sup> <http://www.mysql.com>.

<sup>216</sup> <http://www.openoffice.org>.

<sup>217</sup> <http://www.gimp.org>.

<sup>218</sup> <http://www.perl.org>.

<sup>219</sup> *Jaeger/Metzger*, OSS, S. 16.

<sup>220</sup> Bei Embedded Systemen handelt es sich um Elektronikgeräte mit integriertem Mikroprozessor, wie z.B. Mobiltelefone, Haushaltsgeräte oder Unterhaltungselektronik.

<sup>221</sup> Eine Verpflichtung zur Weiterverbreitung von veränderter Software besteht nicht. Auch hinsichtlich den Bedingungen einer Weiterverbreitung bestehen je nach Lizenzmodell unterschiedliche Anforderungen, vgl. dazu unten IV. 3.

Software bei weitem übersteigt. Um die mit dem Open Source Modell verbundenen Freiheiten für den Nutzer auch rechtlich abzusichern, wurde frühzeitig mit der Entwicklung entsprechender Lizenzen begonnen. Mittlerweile gibt es eine kaum noch zu überblickende Vielzahl an Lizenzen, denen jedoch stets gemein ist, dass die urheberrechtlichen Ausschließlichkeitsrechte dazu verwendet werden, die oben erwähnten Nutzungsmöglichkeiten zu ermöglichen und zu sichern. Grundlage aller Open Source Lizenzen ist dabei die sog. Open Source Definition.

#### **a. Die Open Source Definition**

Die Open Source Definition wurde von *Bruce Perens* von der Open Source Initiative (OSI) entwickelt.<sup>222</sup> Jede Lizenz, die den Richtlinien der Open Source Definition entspricht, darf als Open Source Lizenz bezeichnet werden und kann als solche von der Open Source Initiative zertifiziert werden. Um der Definition als Open Source Software gerecht zu werden, müssen die Lizenzbedingungen einer Software folgende Anforderungen erfüllen:

- Die Lizenz muss die unbeschränkte und lizenzgebührenfreie Weiterverbreitung der Software gestatten.
- Die Software muss zusätzlich im Quellcode vorliegen oder dieser muss jedenfalls offen zugänglich sein, und die Lizenz muss eine Weiterverbreitung des Quellcodes gestatten.
- Die Lizenz muss die Veränderung und die Weiterentwicklung der Software erlauben. Sie muss die Weiterverbreitung der veränderten Software unter den gleichen Lizenzbedingungen gestatten.
- Um es dem Autor eines Programms zu ermöglichen seine Code von anderem Code getrennt zu halten, darf die Verbreitung von modifiziertem Quellcode dann eingeschränkt werden, wenn sie jedenfalls die Verbreitung von Patches in Verbindung mit dem Originalcode gestattet. Es besteht zudem die Möglichkeit den Autor des veränderten Programms zu verpflichten, dieses unter einem anderen Namen oder einer anderen Versionsnummer zu vertreiben.
- Die Lizenz darf keine Personen oder Personengruppen von der Nutzung ausschließen.

---

<sup>222</sup> Siehe <http://www.opensource.org/docs/definition.php>.

- Die Nutzung der Software darf nicht auf bestimmte Einsatzgebiete beschränkt werden. Auch eine kommerzielle Nutzung ist gestattet.
- Die Lizenzbedingungen müssen für jeden gelten, der die Software erhalten hat.
- Die zum Programm gehörigen Rechte dürfen nicht davon abhängen, dass das Programm teil einer bestimmten Softwaredistribution ist.
- Die Lizenz darf keine andere Software beeinträchtigen, die zusammen mit Open Source Software vertrieben wird.
- Die Lizenz muss technologie-neutral sein.

Die Open Source Definition stellt eine Rahmenlizenz dar, an dessen Vorgaben sich jede Open Source Lizenz halten muss. Jedoch lässt die Open Source Definition genügend Spielraum, um eine darauf basierende Lizenz den eigenen Bedürfnissen anzupassen. Es sollte dabei aber betont werden, dass keine Open Source Lizenzart den kommerziellen Vertrieb eines Open Source Programms verbietet darf (vgl. Ziff. 6). Allein die Erhebung von Lizenzgebühren ist ausgeschlossen. Entgelte für Datenträger, Beratung, Garantien oder sonstige Serviceleistungen können dagegen ohne weiteres verlangt werden. Darauf basiert auch das äußerst erfolgreiche Geschäftsmodell sog. GNU/Linux-Distributionen. Open Source Software sollte also keinesfalls mit kostenloser Software gleichgesetzt werden.<sup>223</sup>

### **b. Freie Software versus Open Source Software**

Der Begriff „Open Source Software“ ist noch relativ jung. Er wurde am 3. Februar 1998 bei der Gründung der Open Source Initiative (OSI) in Palo Alto (Kalifornien) geprägt. Davor verwendete man den Begriff „Freie Software“ um dieses Entwicklungsmodell zu beschreiben. Der Begriff „Freie Software“ (engl. free software) wurde 1984 von *Richard Stallmann* geprägt, als das dem GNU-Projekt zugrunde liegende Entwicklungsmodell. Ähnlich wie bei der Open Source Definition stellte auch *Stallmann* mehrere Kriterien auf, anhand derer sich Freie Software von proprietärer Software

---

<sup>223</sup> Trotz allem wird dieses „Argument“ immer wieder von proprietären Softwareunternehmen angeführt, um die Qualität von Open Source Produkten in Frage zu stellen. Etwas was kostenlos sei, könne nicht gleichzeitig von hoher Qualität sein.

unterscheiden sollte. Entscheidend sind dabei vier Grundfreiheiten<sup>224</sup> des Nutzers:

- Die Freiheit, ein Programm für jeden Zweck ohne Einschränkungen benutzen zu dürfen (Freiheit 0).
- Die Freiheit, untersuchen zu dürfen, wie ein Programm funktioniert, und es den eigenen Bedürfnissen anzupassen (Freiheit 1).
- Die Freiheit, ein Programm kopieren zu dürfen und anderen zur Verfügung zu stellen (Freiheit 2).
- Die Freiheit, ein Programm verbessern zu dürfen und diese Verbesserungen der Allgemeinheit zum allgemeinen Wohl zugänglich zu machen (Freiheit 3).

Trotz zahlreicher Bemühungen *Stallmanns*<sup>225</sup> und der FSF, wurde dem Begriff „Freie Software“ aufgrund seiner Mehrdeutigkeit und seiner vermuteten „anti-kommerziellen“ Ausrichtung eher mit zurückhaltender Skepsis oder offenkundiger Ablehnung begegnet. Trotzdem interessierten sich zunehmend auch große Softwareunternehmen wie IBM, Sun oder Netscape für das der Freien Software zugrunde liegende Entwicklungsmodell. Angesichts der kurz bevorstehenden Freigabe des Quellcodes des neuen Communicators 5.0 durch Netscape<sup>226</sup>, sah die Open Source Initiative die Gelegenheit gekommen, die Philosophie der Freien Software durch die Wahl eines unverfänglicheren Namens auch der kommerziellen Softwareindustrie „schmackhaft“ zu machen. Aus diesem Grunde entschied man sich, künftig den Begriff „Open Source Software“ zu verwenden. Die Marketingoffensive der Open Source Initiative hatte vollen Erfolg, wurde jedoch nicht von allen Vertretern der Freien Software Szene befürwortet. Insbesondere *Stallmann* und die FSF lehnten den Begriffswechsel ab.<sup>227</sup> Wie sich jedoch aus dem Vergleich der Open Source Definition und der Free Software Definition ergibt, verbirgt sich hinter beiden Begriffen im Grunde genommen dasselbe Modell. Während der Begriff „Open Source Software“ eine eher pragmatische Einstellung zu dem zugrunde liegenden Lizenzmodell ausdrückt,

---

<sup>224</sup> Vgl. Free Software Definition, <http://www.fsf.org/philosophy/free-sw.html>.

<sup>225</sup> Berühmt wurde *Stallman's* Analogie: “To understand the concept, you should think of free as in “free speech”, not as in “free beer”.

<sup>226</sup> <http://www.netscape.com/newsref/pr/newsrelease558.html>.

<sup>227</sup> <http://www.fsf.org/philosophy/free-software-for-freedom.html>.

versuchen die Verfechter des Begriffs „Freie Software“ mehr die soziale Bedeutung der Lizenz in den Vordergrund zu stellen. Lässt man diese grundlegenden weltanschaulichen Differenzen beiseite, ist unter beiden Bezeichnungen im Ergebnis dasselbe Lizenzmodell zu verstehen.<sup>228</sup> Jedenfalls in der juristischen Literatur hat sich der Begriff „Open Source Software“ weitgehend durchgesetzt<sup>229</sup>, so dass auch im Folgenden an diesem Begriff festgehalten wird.

### c. Open Source Lizenzmodelle

Mittlerweile gibt es eine kaum noch zu überschauende Anzahl von Open Source Lizenzen.<sup>230</sup> Trotz dieser Vielzahl von Lizenzen, haben sich im Prinzip zwei unterschiedliche Grundtypen herausgebildet, die sog. Copyleft-Lizenzen und die sog. Non-Copyleft-Lizenzen.

#### aa. Copyleft Lizenzen

Der Begriff „Copyleft“ wurde von der FSF entwickelt und beschreibt einen zentralen Effekt der GNU GPL und artverwandten Lizenzen. Der Copyleft Effekt beschreibt eine Regelung, die sicherstellen soll, dass ein einmal unter eine Open Source Lizenz gestelltes Produkt auf Dauer auch frei bleibt. Es soll sichergestellt werden, dass Innovationen, die auf der Leistung anderer Open Source Programmierer beruhen, der Gemeinschaft nicht wieder entzogen werden können. Dieses Ergebnis wird dadurch realisiert, dass jede Copyleft Lizenz eine Schutzklausel enthält, die sicherstellt, dass die Weiterverbreitung einer bearbeiteten Version eines Open Source Programms nur unter denselben Bedingungen wie das ursprüngliche Programm möglich ist (sog. viraler Effekt). Jeder Versuch die Weiterverbreitung unter anderen Bedingungen durchzuführen, führt automatisch zum Verlust sämtlicher Rechte unter der ursprünglichen Lizenz. Dadurch ist es nicht möglich ein einmal unter einer Copyleft Lizenz freigegebenes Programm unter restriktiveren oder gar proprietären Bedingungen zu vertreiben.

---

<sup>228</sup> So auch *Moglen*, Free Software or Open Source?, S.1.

<sup>229</sup> *Wuermeling/Theis*, CR 2003, 87ff.; *Omsels*, FS-Hertin, S. 141ff.; *Jaeger/Metzger*, OSS, S. 3f.; *Schiffner*, OSS, S. 4ff.; *Sandl*, CR 2001, 346ff.; *Sester*, CR 2000, 797ff.; *Spindler*, Rechtsfragen der OSS, S. 8ff.; anders hingegen *Grzeszick*, MMR 2000, 412ff.

<sup>230</sup> Eine Übersicht findet sich unter [http://www.ifross.de/ifross\\_html/lizenzcenter.html](http://www.ifross.de/ifross_html/lizenzcenter.html) und <http://www.fsf.org/licenses/license-list.html>.

### **bb. Non-Copyleft Lizenzen**

Bei Non-Copyleft Lizenzen, wie z.B. den BSD-artigen Lizenzen<sup>231</sup>, fehlt es an einer mit der Copyleft-Klausel vergleichbaren Schutzklausel. Bei dieser Open Source Lizenz besteht also die Möglichkeit, ein Programm nachdem es bearbeitet wurde, als proprietäres Produkt zu vertreiben.<sup>232</sup> Die Vereinbarkeit eines solchen Vorgehens mit der Open Source Definition ergibt sich aus Ziff. 1 OSD, wonach die Weiterverbreitung unter den gleichen Lizenzbedingungen zwar gestattet werden muss, nicht jedoch zwingend vorgesehen werden muss. Ein großer Vorteil einer Non-Copyleft Lizenz ist darin zu sehen, dass damit ein proprietäres Programm zusammen mit anderen Open Source Programmen oder Open Source Programmbibliotheken vertrieben werden kann, ohne dass das Unternehmen befürchten muss, durch den Copyleft-Effekt irgendwelche Rechte an seinem Programm einbüßen zu müssen. Häufig sprechen daher wirtschaftliche Gesichtspunkte für die Wahl einer Non-Copyleft Lizenz.

### **cc. Dual Licensing**

Immer beliebter wird auch der Vertrieb eines Programms im Wege des sog. Dual Licensing.<sup>233</sup> Darunter ist der Vertrieb eines Programms unter mehreren, insbesondere unter einer proprietären und einer Open Source Lizenz zu verstehen. Ein Vertrieb im Wege des Dual Licensing ist bei einem Programm, welches unter einer Copyleft Lizenz steht, nur möglich, wenn der Lizenzgeber die ausschließlichen Nutzungsrechte an dem Programm besitzt. Der Vertrieb eines fremden Programms, welches unter einer Copyleft Lizenz steht ist hingegen nicht möglich. Bei Non-Copyleft Lizenzen ist ein Dual Licensing hingegen stets möglich, unabhängig davon, wer die Software ursprünglich entwickelt hat.

---

<sup>231</sup> Berkeley Software Distribution, vgl. dazu *Jaeger/Metzger*, OSS, S. 54ff.

<sup>232</sup> Das ursprüngliche Programm bleibt selbstverständlich als Open Source Programm erhalten.

<sup>233</sup> Dual Licensing wird beispielsweise bei Vertrieb von OpenOffice, MySQL oder Ghostscript verwendet.

## **d. Abgrenzung zu anderen Vertriebsformen**

### **aa. Proprietäre Software**

Das Gegenstück zu Open Source Software ist die sog. proprietäre (herstellerspezifische) Software. Kennzeichnend für proprietäre oder auch „Closed Source Software“ ist zum einen, dass der Quellcode eines Programms gerade nicht offen gelegt wird. Darüber hinaus ist kennzeichnend für proprietäre Software, dass sich der Urheber das exklusive Recht zur Bearbeitung, Vervielfältigung und Weiterverbreitung vorbehält, und der Nutzer das Recht zur Nutzung der Software nur gegen Zahlung einer Lizenzgebühr erhält. Proprietäre Softwarelizenzen sind daher in der Regel so ausgestaltet, dass der Nutzer keine über die bestimmungsgemäße Nutzung des Programms hinausgehende Rechte erhält. Insofern kann proprietäre durchaus als das spiegelbildliche Gegenstück von Open Source Software bezeichnet werden.

### **bb. Freeware**

Auch wenn die Bezeichnung „Freeware“ das Gegenteil vermuten lässt, handelt es sich bei dieser Vertriebsform ebenfalls um eine Erscheinungsform der proprietären Software. Insofern wird Freeware teilweise zu Unrecht mit Free Software oder Open Source Software verwechselt.<sup>234</sup> Zwar sieht auch Freeware eine unentgeltliche Weitergabe der Software vor, im Gegensatz zu Open Source Software ist bei Freeware eine Veränderung des Quellcodes jedoch nicht gestattet und in der Regel auch nicht möglich, da der Quellcode nicht zur Verfügung steht. Freeware ist somit keine freie Software.<sup>235</sup> Zwar gilt bei Freeware das Vervielfältigungs- und Weiterverbreitungsverbot nicht, wohl aber alle anderen einschränkenden Lizenzbedingungen proprietärer Software. Proprietäre Software wird häufig als Freeware vertrieben, um einen strategischen Marktvorteil zu erlangen. Prominente Beispiele dafür sind etwa der Internet Explorer von Microsoft oder der Acrobat Reader von Adobe.

---

<sup>234</sup> So etwa *Gehring*, Freeware, Shareware und PD, Punkt 2.3.; *Spindler*, Rechtsfragen der OSS, S. 18.

<sup>235</sup> So auch *Deike*, CR 2003, 9 (10); *Jaeger/Metzger*, OSS, S. 6; *Schiffner*, OSS, S. 23; vgl. auch die Definition der FSF; <http://www.gnu.org/philosophy/categories.html>.

### **cc. Shareware**

Als Shareware wird diejenige Software bezeichnet, die zwar frei kopiert und verbreitet, aber nur zu einem bestimmten Zweck (beispielsweise für den privaten Einsatz) oder über einem gewissen Zeitraum hinweg kostenlos genutzt werden kann. Nach dieser Testphase muss der Nutzer, falls er die Software weiterhin nutzen möchte, eine Lizenzgebühr an den Autor zahlen. Um der erfahrungsgemäß geringen Zahlungsmoral der Nutzer vorzubeugen, ist Shareware zudem oft so modifiziert, dass sie während der Testzeit nur einen eingeschränkten Funktionsumfang bietet und/oder danach ihren Dienst einstellt. Shareware wird häufig von privaten Programmieren oder kleinen Firmen eingesetzt, um kostengünstig auf die eigene Produktlinie aufmerksam zu machen. Bekannte Beispiele von Shareware sind das Komprimierungsprogramm WinZIP, das Grafikbearbeitungsprogramm Paint Shop Pro oder das IRC Chat Programm mIRC. Auch Shareware stellt daher im Ergebnis eine Form von proprietäre Software dar.

### **dd. Public Domain Software**

Public Domain ist ein Rechtsbegriff aus den USA, welcher sich aufgrund der unterschiedlichen Ausgestaltung des amerikanischen Copyright Systems und des deutschen Urheberrechts in solcher Form im deutschen Recht nicht wieder findet. Public Domain Software bezeichnet Computerprogramme, welche keinem Urheberrecht unterliegen und somit zum Allgemeingut zählen. Ein völliger Verzicht auf das Urheberrecht ist nach deutschem Recht jedoch nicht möglich (vgl. § 29 UrhG). Seinen Ursprung hat dieses Modell in den siebziger Jahren in den USA.<sup>236</sup> Sämtliche mit öffentlichen Mitteln an amerikanischen Universitäten entwickelten Softwareprogramme mussten der Allgemeinheit als Public Domain Software zur Verfügung gestellt werden. Dieses Modells bedienten sich in der Folge auch viele private Autoren von Software. Im Ergebnis ist damit eine uneingeschränkte Vervielfältigung, Weiterentwicklung und Weiterverbreitung von Public Domain Software erlaubt. In dieser Hinsicht besteht eine gewisse Übereinstimmung zu Open Source Software. Ein grundlegender Unterschied zu Open Source Software besteht jedoch darin, dass der Quellcode nicht mit der Software

---

<sup>236</sup> *Siepmann*, JurPC Web-Dok. 163/1999, Abs. 13.

verbreitet werden muss. Außerdem kann ein Public Domain Programm jederzeit in proprietäre Software verwandelt werden.

### **3. Rechtliche Ausgestaltung einer Open Source Lizenz am Beispiel der GNU General Public License**

Die GNU GPL stellt das Urmodell aller Open Source Lizenzen mit Copyleft-Effekt dar. Zudem wird nach wie vor die überwiegende Anzahl aller Open Source Programme unter der GNU GPL lizenziert. Von daher soll die besondere rechtliche Ausgestaltung einer Open Source Lizenz am Beispiel der GNU GPL dargestellt werden. Die vertragsrechtliche und urheberrechtliche Ausgestaltung der GNU GPL ist mittlerweile weitgehend geklärt.<sup>237</sup> Im Folgenden sollen nur jene Bestimmungen dargestellt werden, die auch aus patentrechtlicher Sicht von Bedeutung sein könnten.<sup>238</sup> Der Anwendungsbereich der GNU GPL wird in § 0 GPL geregelt. Danach gilt die GPL nur für die Vervielfältigung, Verbreitung und Bearbeitung eines Programms. Die bloße Nutzung eines Programms wird nicht von der GPL erfasst.<sup>239</sup> Die Rechte des Nutzers sind in §§ 1 und 2 GPL geregelt. Gemäß § 1 GPL hat der Nutzer das Recht, auf beliebigen Medien unveränderte Kopien des Programms herzustellen und zu verbreiten, sofern der Text der GPL jeder Kopie beigefügt wird. Gemäß § 2 GPL hat der Nutzer darüber hinaus auch das Recht, ein Programm zu verändern. Derartige Bearbeitungen dürfen gemäß § 2 GPL aber nur unter der Bedingung vervielfältigt und verbreitet werden, dass die veränderten Dateien mit einem Vermerk über Datum und Art der Bearbeitung versehen sind (§ 2 a) GPL), die Software Dritten gegenüber als Ganzes unter den Bedingungen der GPL und ohne Lizenzgebühren zur Verfügung gestellt wird (§ 2 b) GPL) und die Software eine entsprechende Meldung ausgibt oder ausdrückt (§ 2 c) GPL). Inhaltlich handelt es sich bei den Rechten der §§ 1 und 2 GPL um die Einräumung eines einfachen Nutzungsrechts gemäß § 31 II UrhG.<sup>240</sup> Diese Rechte werden dem Empfänger gemäß § 6 S.1 GPL vom ursprünglichen Lizenzgeber eingeräumt. Die Bestimmungen der GPL stellen dabei Allgemeine Geschäftsbedingungen im Sinne des § 305 I

---

<sup>237</sup> Im Einzelnen dazu: *Jaeger/Metzger*, OSS, S. 31ff. u. 137ff.; *Schiffner*, OSS, S. 147ff.; *Spindler*, Rechtsfragen der OSS, S. 27ff.

<sup>238</sup> Vgl. dazu unten V.2.c.

<sup>239</sup> Dieses Recht ergibt sich ohnehin schon aus § 69d UrhG.

<sup>240</sup> *Jaeger/Metzger*, OSS, S. 32; *Omsels, FS-Hertin*, S. 155; *Schiffner*, OSS, S. 154f.; *Spindler*, Rechtsfragen der OSS, S. 27.

1 BGB dar.<sup>241</sup> Wann eine Einbeziehung in den Vertrag erfolgt, wird unterschiedlich beurteilt. Zum Teil wird eine Einbeziehung bereits beim Download bzw. bei der Installation des Programms angenommen.<sup>242</sup> Nach anderer Ansicht kommt es erst mit der Veränderung der Software eines Open Source Programms zu einer Einbeziehung der Bestimmungen.<sup>243</sup> Für die letztgenannte Ansicht spricht auch der Wortlaut des § 5 S.3 GPL. Auf den Zugang der Annahmeerklärung wird gemäß § 151 S.1 BGB verzichtet.<sup>244</sup> Fraglich ist allerdings, welche rechtlichen Auswirkungen ein Verstoß gegen die Bestimmungen der GPL hat. Gemäß § 4 S.2 GPL führt jeder Verstoß gegen die Bestimmungen der GPL automatisch zum Erlöschen sämtlicher nach der GPL gewährten Rechte. Nach der überwiegenden Ansicht erfolgt die Einräumung des einfachen Nutzungsrechts damit unter einer auflösenden Bedingung im Sinne des § 158 II BGB.<sup>245</sup> Jede Versuch, ein unter der GPL lizenziertes Programm nach einer Bearbeitung unter anderen (z.B. proprietären) Lizenzbedingungen zu verbreiten, führt also automatisch zum Erlöschen des einfachen Nutzungsrechts an dem Programm, mit der Folge, dass das Programm nicht vertrieben werden darf. Damit wird gewährleistet, dass eine Copyleft-Software nicht in eine proprietäre Software umgewandelt werden kann. Inwiefern sich dieser Schutzmechanismus auch auf Patente erstreckt, wird noch zu klären sein.<sup>246</sup>

## V. Das Spannungsverhältnis von Open Source Software und Softwarepatenten

*„Schließlich und endlich ist jedes freie Programm permanent durch Software-Patente bedroht.“*

Dieser Satz, welcher sich seit Version 2 von 1991 in der Präambel der GNU GPL findet, spiegelt die kritische Haltung der Open Source Bewegung gegenüber der Patentierung von Software wider. In

---

<sup>241</sup> Deike, CR 2003, 9 (13); Jaeger/Metzger, OSS, S. 147; Koch, CR 2000, 333 (339); Omsels, FS-Hertin, S. 147ff.; Schiffner, OSS, S. 174; Sester, CR 2000, 797 (804).

<sup>242</sup> So etwa Deike, CR 2003, 9 (13).

<sup>243</sup> Jaeger/Metzger, OSS, S. 148.

<sup>244</sup> Deike, CR 2003, 9 (13); Jaeger/Metzger, OSS, S. 149; Sester, CR 2000, 797 (804).

<sup>245</sup> Deike, CR 2003, 9 (16); Grzeszick, MMR 2000, 413 (415); Metzger/Jaeger, GRUR Int. 1999, 839 (843); Sester, CR 2000, 797; Spindler, Rechtsfragen der OSS, S. 31; a.A. Schiffner, OSS, S. 164, der von der Einräumung eines inhaltlich beschränkten einfachen Nutzungsrechts gemäß §§ 31 I, 32 UrhG ausgeht.

<sup>246</sup> Siehe unten V.2.c.

diesem Abschnitt soll untersucht werden, ob und inwiefern Softwarepatente eine Gefahr für die Entwicklung, den Vertrieb und die Nutzung von Open Source Produkten darstellen. In einem zweiten Schritt gilt es zu erörtern, ob es nach der geltenden Rechtslage in Deutschland möglich ist, eventuell vorhandene Risiken zu vermeiden oder wenigstens einzudämmen. Zunächst erscheint es jedoch angebracht einen kurzen Überblick über die allgemeine rechtspolitische Diskussion betreffend der Patentierung von Software zu geben, die sich keineswegs nur auf das Spannungsverhältnis zwischen Open Source Software und Softwarepatenten beschränkt.

### **1. Einführung in die allgemeine rechtspolitische Diskussion um Softwarepatente**

Die grundlegende Diskussion um Softwarepatente, welche mit wechselnder Vehemenz seit den Anfangstagen der Kommerzialisierung von Software geführt wird, befasst sich mit der fundamentalen Frage, ob ein patentrechtlicher Schutz für Software aufgrund der besonderen Charakteristika der Softwareindustrie überhaupt mit Sinn und Zweck des Patentrechts zu vereinbaren ist. Ziel des Patentrechts ist seit jeher die Förderung des technischen Fortschritts. Durch die Gewährung eines zeitlich beschränkten Ausschließlichkeitsrechts soll ein Entwickler für seine oft mühevollen und kostspieligen Forschungs- und Entwicklungsarbeiten belohnt werden und so zur Offenbarung seiner Erfindung angeregt werden.<sup>247</sup> Dass Patente z.B. im Bereich der pharmazeutischen Industrie, des Maschinenbaus oder der Luft-, Raum- und Fahrzeugtechnik zweifelsohne eine wichtige Rolle bei der Forschung und der Entwicklung neuer Produkte gespielt haben, kann nicht ernsthaft bestritten werden. Ob dies generell für jede Industrie gelten muss, ist damit aber noch nicht beantwortet. Die Gegner von Softwarepatenten jedenfalls bezweifeln dies in Bezug auf die Softwareindustrie. Inwiefern Softwarepatente tatsächlich negative Auswirkungen auf Forschung und Innovationen im Bereich der Softwareentwicklung haben, ist nicht abschließend geklärt.<sup>248</sup> Neuere Studien deuten zwar darauf hin, dass die besonderen Charakteristika der Softwareindustrie, insbesondere die weite Verbreitung von kleinen

---

<sup>247</sup> Benkhard-Bruchhausen, PatG, Einleitung, Rn. 1.

<sup>248</sup> Vgl. die Ergebnisse von Nack/Straus, MPI-Gutachten, S. 223ff.; Lutterbeck/Gehring/Horns, Kurzgutachten, S. 13ff.

und mittleren Unternehmen, die sequenzielle und komplementäre Natur des Entwicklungsprozesses von Software und die relativ kurzen Innovations- und Produktzyklen der Sinnhaftigkeit eines Patentschutzes entgegen stehen könnten.<sup>249</sup> Es soll nicht Aufgabe dieser Arbeit sein, die mikro- und makroökonomischen Auswirkungen von Softwarepatenten auf die gesamte Softwareindustrie zu untersuchen. Auszugehen ist hier vielmehr von der Tatsache, dass die Gewährung von Patenten für computerbezogene Erfindungen Realität ist. In Anbetracht der europäischen und internationalen Verflechtungen – zu denken ist hier nur an das EPÜ, die geplante EU-Richtlinie zur Patentierung von computer-implementierten Erfindungen oder an das TRIPS Übereinkommen – kann auch nicht davon ausgegangen werden, dass es in Deutschland in absehbarer Zeit zu einer wesentlichen Änderung dieser Praxis kommen wird. Zu klären ist daher, welche Gefahren mit der Patentierbarkeit von Software für das Open Source Software Entwicklungsmodell verbunden sind und welche Möglichkeiten bestehen, mit diesen Gefahren in adäquater Art und Weise umzugehen.

## **2. Spezifische Gefahren für Entwicklung, Vertrieb und Nutzung von Open Source Software durch Softwarepatente**

Die Gefahren für die Entwicklung, den Vertrieb und die Nutzung von Open Source Software durch Softwarepatente lassen sich in drei Gruppen einteilen. Zum einen besteht die Gefahr, dass bei Entwicklung, Vertrieb oder Nutzung eines Open Source Produkts Patente proprietärer Software verletzt werden. Des weiteren besteht die Gefahr, dass Eigenentwicklungen der Open Source Bewegung durch Patentanmeldungen seitens der Entwickler proprietärer Software monopolisiert werden. Ein dritter Gefahrenherd ist schließlich in der Anmeldung von Patenten aus den eigenen Reihen der Open Source Bewegung zu erblicken.

### **a. Verletzung von Patenten proprietärer Software**

Eine wesentliche Gefahr, die bei der Entwicklung, beim Vertrieb und bei der Nutzung von Open Source Software besteht, ist in der Möglichkeit der Verletzung existierender Patente proprietärer

---

<sup>249</sup> Vgl. *Bessen/Maskin*, Sequential Innovation, Patents and Imitation, 2000, <http://www.researchoninnovation.org/patent.pdf>; *Marly*, Computersoftware in der EU, S. 51f.

Software zu sehen. Dass es sich hierbei nicht nur um ein theoretisches, sondern um sehr reales Problem handelt, zeigen folgende Beispiele: Aufgrund eines Patents der Firma UNISYS auf das LZW-Kompressionsverfahren<sup>250</sup>, welches wiederum Bestandteil der populären GIF- und TIFF-Grafikformate ist, unterstützt das Open Source Grafikbearbeitungsprogramm GIMP standardmäßig das Laden und Abspeichern von GIF und TIFF Grafiken nicht.<sup>251</sup> Auch das CMYK-Farbmodell wird von GIMP (noch) nicht unterstützt, weil die Firma Adobe darauf ein Patent hält. Dadurch scheidet GIMP für die professionelle Bildbearbeitung aus. Schriften werden mit XFree86 teilweise recht langsam und unansehnlich wiedergegeben, weil mehrere Routinen des TrueType Fontformats von Apple patentiert wurden.<sup>252</sup> Der Distributor RedHat wiederum hat sämtliche MP3 Software aus seiner Distribution entfernt, um potentiellen Konflikten mit den Patentinhabern Fraunhofer Institut und Thompson Multimedia aus dem Weg zu gehen. Die obigen Beispiele stellen nur einen kleinen Ausschnitt aus dem Dickicht patentierter computerbezogener Erfindungen dar. Die Gefahr bei der Entwicklung von Open Source Software ein bereits existierendes proprietäres Softwarepatent zu verletzen, ist daher durchaus als real bewerten. Zwar stellen sich diese Probleme auch den Entwicklern von proprietärer Software, zu bedenken ist aber, dass an der Entwicklung von Open Source Software in der Regel eine Vielzahl von Einzelpersonen oder jedenfalls eher kleine Softwarefirmen beteiligt sind, die weder über die personellen noch über die finanziellen Mittel verfügen, um umfangreiche Patentrecherchen durchzuführen. Zudem wird die Feststellung einer Patentverletzung durch den offenen Quellcode von Open Source Software erheblich erleichtert. Besonders in den USA wird die Situation dadurch verschärft, dass gerade im Bereich der computerbezogenen Erfindungen in erhöhtem Maße sog. Trivialpatente erteilt wurden und nach wie vor erteilt werden.<sup>253</sup>

---

<sup>250</sup> Das Patent ist in den USA bereits abgelaufen und läuft in Deutschland noch bis zum 18.06.2004; vgl. [http://www.unisys.com/about\\_\\_unisys/lzw/](http://www.unisys.com/about__unisys/lzw/).

<sup>251</sup> Zur Vertiefung siehe <http://burnallgifs.org/>.

<sup>252</sup> Vgl. dazu <http://www.freetype.org/patents.html>.

<sup>253</sup> Eine Auflistung fragwürdiger Softwarepatenten findet sich unter: [http://www.oreillynet.com/pub/q/patent\\_list](http://www.oreillynet.com/pub/q/patent_list).

### aa. Verletzung von Erzeugnis- und Verfahrenspatenten

Der vom Patentschutz erfasste Gegenstand wird gemäß § 14 PatG durch den Inhalt der Patentansprüche definiert. Eine Verletzung kann sich als eine identische Benutzung der beanspruchten Erfindung oder als eine inhaltsgleiche Verwirklichung des Erfindungsgegenstandes darstellen.<sup>254</sup> Eine identische Benutzung ist gegeben, wenn die angegriffene Ausführungsform sämtliche Lösungsmerkmale der angemeldeten Lehre verwirklicht. Eine inhaltsgleiche Benutzung liegt vor, wenn Merkmale des Patentanspruchs durch in ihrer technischen Funktion übereinstimmende Lösungsmittel mit gleicher Wirkung ausgetauscht werden.<sup>255</sup> Eine Ausführungsform, die einen anderen als den vom Patent vorgeschlagenen Lösungsweg verfolgt und nicht alle anspruchsgemäßen Merkmale in identischer oder äquivalenter Form aufweist, fällt hingegen nicht unter den Schutzbereich des Patents.<sup>256</sup>

Ob eine Patentverletzung vorliegt hängt entscheidend davon ab, ob es sich bei dem angemeldeten Patent um ein Erzeugnis- oder ein Verfahrenspatent handelt. Bei Erzeugnispatenten im Sinne des § 9 Nr. 1 PatG ist Gegenstand des Patents eine konkrete Vorrichtung wie z.B. eine Maschine, ein Apparat oder ein Produkt. Eine Verletzung liegt vor, wenn ein Erzeugnis, das Gegenstand des Patents ist, unberechtigt hergestellt, angeboten, in Verkehr gebracht oder gebraucht wird. Ist Gegenstand des Patents eine Vorrichtung in Form eines Computersystems mitsamt Software, so liegt eine Verletzung im Sinne des § 9 Nr. 1 PatG nur dann vor, wenn die ihrer Funktion nach identische Open Source Software ebenfalls in Kombination mit der beanspruchten Hardware hergestellt, angeboten, in Verkehr gebracht oder gebraucht wird. Eine derartige Verletzung kommt insbesondere bei „Embedded Systemen“ in Betracht.<sup>257</sup> In der Regel wird eine Software jedoch isoliert von einer konkreten Hardwarevorrichtung angeboten. In diesem Fall kommt allenfalls eine mittelbare Patentverletzung gemäß § 10 I PatG in Betracht. Aufgrund der strengeren subjektiven Voraussetzungen eines Verstoßes gegen § 10 I PatG, dürfte sich ein darauf gestützter Unterlassungsanspruch

---

<sup>254</sup> *Benkhard-Ullmann*, PatG, § 14, Rn. 117.

<sup>255</sup> *Benkhard-Ullmann*, PatG, § 14, Rn. 123.

<sup>256</sup> *Benkhard-Ullmann*, PatG, § 14, Rn. 116; *Busse-Keukenschrijver*, PatG, § 14, Rn. 82; *Heide*, CR 2003, 165 (166); *Kraßer* in: *Lehmann*, Rechtsschutz von Computerprogrammen, S. 275, Rn. 118.

<sup>257</sup> *Heide*, CR 2003, 165 (166).

nur schwer durchsetzen lassen.<sup>258</sup> Anders ist die Situation freilich dann zu beurteilen, wenn sich der Patentanspruch auf ein „Computerprogrammprodukt“ an sich erstreckt.<sup>259</sup> Hier stellt bereits das Herstellen, Anbieten, In-Verkehr-Bringen, oder Gebrauchen der Software selbst eine Patentverletzung dar. Handelt es sich bei dem Patentanspruch um einen Verfahrensanspruch, so liegt gemäß § 9 Nr. 2 PatG eine Verletzung vor, wenn ein Verfahren, das Gegenstand des Patents ist, angewendet oder zur Anwendung angeboten wird. Im Falle des Anbietens liegt eine Verletzung aber nur dann vor, wenn die Anwendung selbst rechtswidrig ist und dies dem Anbieter bekannt ist oder aufgrund der Umstände offensichtlich ist. Keine Verletzung wäre danach in einem Anbieten zu rein privaten Zwecken (§ 11 Nr. 1 PatG) oder zu Versuchszwecken (§ 11 Nr. 2 PatG) zu sehen. Diese Einschränkung dürfte für den Bereich der Open Source Software jedoch ohne Bedeutung sein, da es gerade kennzeichnend für eine Open Source Lizenz ist, dass dem Nutzer keine Beschränkung hinsichtlich der Nutzung eines Programms auferlegt werden darf (vgl. Ziff. 6 der Open Source Definition).<sup>260</sup>

#### **bb. Ausnahmeregelung des § 11 Nr. 1 und Nr. 2 PatG**

Gemäß § 11 Nr. 1 PatG erstreckt sich die Wirkung eines Patents nicht auf Handlungen, die im privaten Bereich zu nichtgewerblichen Zwecken vorgenommen werden. Eine Handlung im privaten Bereich liegt vor, wenn eine Erfindung für den persönlichen oder häuslichen Bereich oder zu privaten Studienzwecken genutzt wird.<sup>261</sup> Handlungen außerhalb des privaten Bereichs sind unzulässig, unabhängig davon, ob sie zu gewerblichen oder nicht gewerblichen Zwecken vorgenommen werden.<sup>262</sup> Bezogen auf die Entwicklung und die Nutzung von Open Source Software bedeutet dies, dass der Programmierer, der zu Hause an einem Open Source Projekt arbeitet nicht auf einen eventuellen Patentschutz achten muss. Auch der private Anwender, der ein Open Source Produkt zu privaten Zwecken einsetzt, kann dieses lizenzgebührenfrei nutzen. Sobald ein

<sup>258</sup> *Esslinger/Betten*, CR 2000, 18 (20); *Jaeger/Metzger*, OSS, S. 119; a.A. *Heide*, CR 2003, 166 (165).

<sup>259</sup> Die Zulässigkeit eines solchen Patentanspruchs wurde vom EPA, CR 2001, 91ff. – „Computerprogrammprodukt/IBM“ und vom BGH CR 2002, 88 (89) – „Suche fehlerhafte Zeichenketten“ bejaht; vgl. auch Art. 5 Nr. 2 des Vorschlags einer EU-Richtlinie zur Patentierbarkeit von computer-implementierten Erfindungen .

<sup>260</sup> *Jaeger/Metzger*, OSS, S. 119.

<sup>261</sup> *Benkhard-Ullmann*, PatG, § 11, Rn. 3; *Marly*, Computersoftware in der EU, S. 54.

<sup>262</sup> *Benkhard-Ullmann*, PatG, § 11, Rn. 4.

Entwickler ein Programm aber über den privaten Bereich hinaus anbietet, etwa durch das Bereitstellen im Internet, greift § 11 Nr. 1 PatG nicht mehr ein. Dabei spielt es auch keine Rolle, ob das Programm kostenlos und ohne Gewinnerzielungsabsicht angeboten wird. Auch die kommerzielle Nutzung von Open Source Software oder die Nutzung durch öffentliche Institutionen wird nicht von der Ausnahmeregelung des § 11 Nr. 1 PatG erfasst. Die Ausnahmeregelung des § 11 Nr. 1 PatG hat daher für den Bereich der Open Source Software nur einen sehr schmalen Anwendungsbereich.<sup>263</sup> Von Bedeutung dürfte allenfalls die Nutzung durch private Anwender zu privaten Zwecken sein. Gleiches gilt auch für die Ausnahmeregelung des § 11 Nr. 2 PatG. Danach sind Handlungen zu Versuchszwecken nicht vom Schutzbereich des Patents erfasst. Auch hier dürfte sich für die Entwicklung von Open Source Software nur ein enger Anwendungsbereich ergeben, zumal § 69e UrhG eine Dekompilierung nur in engen Grenzen gestattet.

### **cc. Rechtsfolgen**

Bei Vorliegen einer Verletzung eines proprietären Softwarepatents, steht dem Patentinhaber gemäß § 139 I PatG ein Anspruch auf Unterlassung zu. Darüber hinaus besteht gemäß § 139 II PatG ein Anspruch auf Schadensersatz. Dieser setzt jedoch voraus, dass der Patentverletzer vorsätzlich oder fahrlässig handelte. Dem „einfachen“ Open Source Entwickler dürfte in der Regel keine Sorgfaltspflichtverletzung vorzuwerfen sein, wenn er auf umfangreiche Patentrecherchen verzichtet und so in Unkenntnis eines Patents eine Patentverletzung begeht.<sup>264</sup> Etwas anderes gilt jedoch dann, wenn er durch entsprechende Abmahnungen des Patentinhabers auf das Bestehen eines Patents aufmerksam gemacht wurde. Anders ist die Situation jedoch für große Unternehmen zu beurteilen, die sich mit der Entwicklung oder mit der Distribution von Open Source Software befassen. Hier kann bei mangelnden Patentrecherchen durchaus eine Sorgfaltspflichtverletzung zu bejahen sein.<sup>265</sup> Der Schadensersatzanspruch umfasst dann auch einen entgangenen Gewinn gemäß § 252 BGB, welcher im Wege der Lizenzanalogie berechnet wird. Gemäß § 140a PatG steht dem

---

<sup>263</sup> Lutterbeck/Gehring/Horns, Kurzgutachten, S. 100; Jaeger/Metzger, OSS, S. 119.

<sup>264</sup> Jaeger/Metzger, OSS, S. 120.

<sup>265</sup> Jaeger/Metzger, OSS, S. 120.

Patentinhaber unter bestimmten Umständen auch ein Anspruch auf Vernichtung des patentverletzenden Erzeugnisses zu. Verletzt ein Open Source Produkt ein noch nicht erteiltes, aber bereits angemeldetes Patent, so steht dem Patentinhaber gemäß § 33 PatG ein Entschädigungsanspruch zu, sofern der Verletzer wusste oder wissen musste, dass die Erfindung Gegenstand einer Patentanmeldung ist. Zusammenfassend ist festzustellen, dass bei der Entwicklung, dem Vertrieb und der Nutzung von Open Source Software durchaus die Gefahr einer Patentverletzung besteht. Ausgenommen ist lediglich die Nutzung zu rein privaten Zwecken. Schadensersatzansprüche dürften jedoch nur gegenüber größeren Unternehmen mit entsprechenden Möglichkeiten zur Patentrecherche bestehen.

#### **b. Unberechtigte Patentanmeldungen für proprietäre Software**

Eine weitere Gefahr für die Entwickler von Open Source Software ist in unberechtigten Patentanmeldungen für proprietäre Software zu erblicken. Es stellt sich die Frage, wie sich die Entwickler von Open Source Software gegen unberechtigte Patentanmeldungen schützen können.

##### **aa. Veröffentlichung von Eigenentwicklungen**

Wie bereits dargestellt wurde, werden Patente gemäß § 1 I PatG nur für neue Erfindungen erteilt. Neu ist eine Erfindung gemäß § 3 I S. 1 PatG dann, wenn sie nicht zum Stand der Technik zählt. Nach dem sog. objektiven Neuheitsbegriff umfasst der Stand der Technik gemäß § 3 I S. 2 PatG alle Kenntnisse, die vor dem für den Zeitrang der Anmeldung maßgeblichen Tag durch schriftliche oder mündliche Beschreibung, durch Benutzung oder in sonstiger Weise irgendwo auf der Welt der Öffentlichkeit zugänglich gemacht worden sind. Ein effektiver Schutz gegen unberechtigte Patentanmeldungen für proprietäre Software stellt daher eine Veröffentlichung der Eigenentwicklungen durch die Entwickler eines Open Source Programms dar, um diese damit dem Stand der Technik zuzuführen.<sup>266</sup> Auf die Art und Weise der Veröffentlichung kommt es

---

<sup>266</sup> So auch *Cave*, Patents are your friends, [http://dir.salon.com/tech/feature/2001/03/21/open\\_source\\_patents/index.html](http://dir.salon.com/tech/feature/2001/03/21/open_source_patents/index.html); *Esslinger/Betten*, CR 2000, 18 (21); *Jaeger/Metzger*, OSS, S. 114f.; *Perens*, Preparing for the Intellectual Property Offensive, <http://www.linuxworld.com/linuxworld/lw-1998-11/lw-11-the-source-p2.html>.

dabei nicht an. Da sich die Entwicklung von Open Source Software ohnehin überwiegend über das Internet abspielt, bietet sich jedoch eine Veröffentlichung über dieses Medium an. Der quelloffene Charakter von Open Source Entwicklungen stellt sich hier als entscheidender Vorteil heraus, da so die in einem Programm verkörperte technische Lehre leicht zugänglich und nachvollziehbar ist. Ein Nachteil des Internets ist jedoch in dessen dynamischer Natur zu erblicken. So ist es häufig nicht eindeutig nachprüfbar, ab welchem Zeitpunkt eine bestimmte Information im Netz abrufbar war. Da es in Bezug auf die Neuheit aber entscheidend darauf ankommt, dass die offenbarte Lehre vor dem Tag der Anmeldung veröffentlicht war, muss gewährleistet werden, dass der Zeitpunkt einer im Internet veröffentlichten Open Source Entwicklung unzweifelhaft feststeht. Ein zweiter wichtiger Aspekt ist in der Recherchierbarkeit der veröffentlichten Informationen zu sehen. Je leichter die Information zugänglich ist, desto höher ist die Wahrscheinlichkeit, dass sie bei der Recherche nach dem Stand der Technik durch das Patentamt auch berücksichtigt wird. Diese Vorgaben lassen sich am wirkungsvollsten durch die Einrichtung einer zentralen Dokumentationsstelle mit einheitlicher Veröffentlichungsstruktur verwirklichen.<sup>267</sup> Entsprechende Dokumentationsstellen existieren auch bereits. So besteht auf der Seite des Anbieters *IP.com* gegen Entgelt die Möglichkeit Eigenentwicklungen in eine sog. PriorArt Datenbank einzutragen.<sup>268</sup> Die Daten werden mit einem Zeitstempel versehen und zwecks einfacher Recherchierbarkeit einheitlich strukturiert und aufbereitet. Nach Angaben von *IP.com* wird die Datenbank regelmäßig von US-amerikanischen Patentämtern und dem EPA konsultiert. Auch der „Förderverein für eine Freie Informationelle Infrastruktur“ (FFII) bietet auf seiner Internetseite ein Dokumentationssystem an, bei dem Informationen über Open Source Entwicklungen mit einem Zeitstempel versehen werden und in einer öffentlichen Datenbank abgelegt werden können.<sup>269</sup> Zwar können Open Source Entwickler ihre Entwicklungen hier kostenlos archivieren, allerdings weist die Datenbank der FFII noch einige Schwächen in der Darbietung der abgelegten Informationen auf. Idealerweise sollten die Patentämter von vorn herein in den

---

<sup>267</sup> Heide, CR 2003, 165 (171); Jaeger/Metzger, OSS, S. 115f.

<sup>268</sup> <http://www.ip.com/index.html>.

<sup>269</sup> <http://swpat.ffii.org/purci/index.de.html>.

Veröffentlichungsprozess mit einbezogen werden. Eine frühstmögliche und lückenlose Dokumentation von Open Source Entwicklungen stellt daher einen effektiven vorbeugenden Schutz gegen unberechtigte Patentanmeldungen für proprietäre Software dar.

#### **bb. Einspruch und Nichtigkeitsklage**

Da gerade im Bereich der Patentierung von computerbezogenen Erfindungen erhebliche Rechterschwierigkeiten bestehen<sup>270</sup>, kann es häufig vorkommen, dass trotz Vorveröffentlichung einer Open Source Entwicklung ein Patent für eine proprietäre Software erteilt wird. Auch in diesem Fall stehen die Entwickler von Open Source Software jedoch nicht schutzlos dar. So besteht gemäß § 59 I i.V.m. § 21 I Nr. 1 PatG innerhalb von drei Monaten nach Veröffentlichung des Patents die Möglichkeit beim Patentamt Einspruch gegen die Erteilung des Patents zu erheben. Einspruchsberechtigt ist grundsätzlich jedermann<sup>271</sup>, so dass es nicht darauf ankommt, wer als Erfinder des Open Source Programms anzusehen ist. Im Einspruchsverfahren kann die Neuheit der angemeldeten Erfindung dann durch Darbietung der vorveröffentlichten Informationen widerlegt werden. Kommt das Patentamt zu dem Ergebnis, dass es an der erforderlichen Neuheit fehlt, wird das Patent gemäß § 61 I PatG widerrufen. Als weiterer Rechtsbehelf kommt die Erhebung einer Nichtigkeitsklage gemäß § 81 I i.V.m. §§ 22, 21 I Nr.1 PatG in Betracht. Die Nichtigkeitsklage kann gemäß § 81 II PatG nach Ablauf der Einspruchsfrist gemäß § 59 I PatG erhoben werden. Wie sich aus dem Umkehrschluss aus § 81 III PatG („nur der Verletzte“) ergibt, kann auch die Nichtigkeitsklage grundsätzlich von jedermann erhoben werden. Wurde der wesentliche Inhalt eines Patents für eine proprietäre Softwareerfindung ohne Einwilligung des Erfinders den Unterlagen oder Verlautbarungen eines Open Source Projekts entnommen (sog. widerrechtliche Entnahme), so kann der Erfinder oder sein Rechtsnachfolger entweder Einspruch gemäß § 59 I 1 i.V.m. § 21 I Nr. 3 PatG bzw. Nichtigkeitsklage gemäß § 81 I i.V.m. §§ 22, 21 I Nr. 3 PatG erheben oder gemäß § 8 PatG Klage auf Übertragung des Anspruchs auf Erteilung des Patents oder des bereits erteilten Patents erheben.

---

<sup>270</sup> Siehe dazu unten V.3.a.aa.

<sup>271</sup> *Benkhard-Schäfer*, PatG, § 59, Rn. 5; *Mes*, PatG, § 59, Rn. 9.

### **cc. Geltendmachung eines Vorbenutzungsrechts gemäß § 12 I PatG**

In den Fällen, in denen keine Vorveröffentlichung eines Open Source Programms stattgefunden hat, bleibt noch die Möglichkeit der Geltendmachung eines Vorbenutzungsrechts gemäß § 12 I PatG.<sup>272</sup> Gemäß § 12 I PatG tritt die Wirkung eines Patents gegen den nicht ein, der zur Zeit der Anmeldung die Erfindung bereits in Benutzung genommen oder die dazu erforderlichen Veranstaltungen getroffen hatte. Die Reichweite des § 12 I PatG erstreckt sich jedoch nur auf denjenigen, der für seine eigenen Bedürfnisse die in der angemeldeten Lehre enthaltene Erfindung bereits vor der Anmeldung in einer in §§ 9 und 10 PatG aufgeführten Benutzungsarten benutzt hat oder dazu bereits Vorkehrungen getroffen hat. Das Vorbenutzungsrecht kann der Begünstigte auch nur im Wege der Vererbung oder der Veräußerung seines Betriebes an Dritte übertragen. Eine eigene Lizenzierung ist ausgeschlossen.<sup>273</sup> Wesentlich für die Geltendmachung eines Vorbenutzungsrechts ist jedoch auch hier wieder eine lückenlose und substantiierte Dokumentation über den Umfang und den Anfangszeitpunkt der Vorbenutzung.

### **c. Anmeldung von Patenten auf Open Source Software zwecks proprietärer Vermarktung**

Ein weiteres Problem, welches sich in Bezug auf Open Source Software und der Patentierung von computerbezogenen Erfindungen stellt, sind Patentanmeldungen aus den eigenen Reihen zwecks proprietärer Vermarktung. Grundsätzlich erwirbt ein Erfinder, dessen Erfindung aus der Weiterentwicklung eines Open Source Programms entstanden ist ein Recht auf das Patent gemäß § 6 S.1 PatG. Solange keine gemeinschaftliche Erfindertätigkeit im Sinne des § 6 S.2 PatG vorliegt, steht dem Erfinder das Recht auf das Patent alleine zu. Die Tatsache, dass er sich dabei unter Umständen auf die Leistung anderer stützt steht dem Erwerb eines Rechts auf das Patent nicht entgegen. Dies ergibt sich bereits aus § 11 Nr. 2 PatG und basiert auf der Tatsache, dass im Prinzip jede Erfindung auch auf den Leistungen und Errungenschaften anderer aufbaut. Da es sich bei dem

---

<sup>272</sup> So auch *Esslinger/Betten*, CR 2000, 18 (21); *Heide*, CR 2003, 165 (171); *Jaeger/Metzger*, OSS, S. 117.

<sup>273</sup> *Benkhard-Bruchhausen*, PatG, § 12, Rn. 25.

Erfindungsakt auch nicht um ein Rechtsgeschäft sondern um einen Realakt handelt, kann einem Erfinder der Erwerb eines Rechts auf das Patent auch nicht verboten werden. Die Anmeldung von Patenten auf Open Source Programme aus den eigenen Reihen ist daher ohne weiteres möglich. Dass es sich auch dabei nicht nur um eine hypothetische Gefahr handelt, zeigt der Fall um das am 30. November 1999 erteilte Patent<sup>274</sup> für RTLinux, ein echtzeitfähiges GNU/Linux Derivat.<sup>275</sup>

Es fragt sich allerdings, ob mit der Möglichkeit auf Open Source Programme ein Patent anzumelden, auch zugleich die proprietäre Nutzung eines solchen Patents möglich ist. Zu unterscheiden ist dabei zwischen Software, welche unter einer Copyleft Lizenz vertrieben wird und solcher, die unter einer Non-Copyleft Lizenz vertrieben wird. Wie bereits oben festgestellt wurde, gestatten Non-Copyleft Lizenzen ohne weiteres die proprietäre Nutzung einer aus einem Open Source Programm entstandenen Weiterentwicklung.<sup>276</sup> Dementsprechend ist auch die proprietäre Nutzung eines auf einer solchen Weiterentwicklung angemeldeten Patents zulässig. Auch einer dualen Lizenzierung des Patents steht prinzipiell nichts im Wege. Anders könnte die Situation jedoch bei Programmen zu beurteilen sein, die unter einer Copyleft Lizenz vertrieben werden. In der Präambel der GNU GPL heißt es dazu: „Wir möchten die Gefahr ausschließen, dass Distributoren eines freien Programms individuell Patente lizenzieren – mit dem Ergebnis, dass das Programm proprietär würde. Um dies zu verhindern, haben wir klargestellt, dass jedes Patent entweder für freie Benutzung durch jedermann lizenziert werden muss oder überhaupt nicht lizenziert werden darf.“ Die GPL möchte also die proprietäre Nutzung von Softwarepatenten verhindern. Eine spezielle Regelung findet sich dazu in der GPL allerdings nicht. Es kann also nur auf die allgemeine Copyleft-Klausel des §§ 2 b) i.Vm. 4 GPL zurückgegriffen werden. Wie bereits oben festgestellt wurde, besteht die Wirkung der Copyleft-Klausel darin, dass jede Weiterentwicklung oder Bearbeitung eines Open Source Programms nur unter den ursprünglichen Bedingungen der jeweiligen Lizenz weiterverbreitet

---

<sup>274</sup> US Patent Nr. 5.995.745.

<sup>275</sup> <http://www.fsmlabs.com/>.

<sup>276</sup> Siehe oben IV.2.c.bb.

werden darf.<sup>277</sup> Dieser Effekt wird dadurch erzielt, dass der ursprüngliche Urheber die Einräumung der Nutzungsrechte unter die auflösende Bedingung i.S.d. § 158 II BGB stellt, dass der Bearbeiter seinen Verpflichtungen aus der Lizenz nachkommt.<sup>278</sup> Die Wirkung der Copyleft-Klausel ist damit untrennbar mit den urheberrechtlichen Befugnissen des ursprünglichen Urhebers eines Programms verknüpft. Hieraus ergibt sich zugleich, dass die Copyleft-Klausel nicht ohne weiteres auch für Patente Geltung beanspruchen kann. Ein Patent wird nämlich nicht für das urheberrechtlich geschützte linguistische Konstrukt eines Computerprogramms erteilt, sondern für die davon losgelöste Funktionalität, sofern sie sich als eine patentierbare Erfindung darstellt. Daraus ergibt sich, dass die Copyleft-Klausel die proprietäre Nutzung eines Softwarepatents grundsätzlich nicht verhindern kann. Dies gilt aber nur, sofern das im Patent beschriebene Verfahren nicht auf Code aus dem ursprünglichen Programm zurückgreift. Wird das Verfahren durch ein Programm ausgeführt, welches noch Code aus dem ursprünglichen Open Source Programm enthält, so kann die proprietäre Nutzung dieser konkreten Umsetzung des Patents verhindert werden. Mit anderen Worten: Die Anmeldung eines Patents kann ein Open Source Programm nicht in ein proprietäres Programm verwandeln, denn hier greift der durch das Urheberrecht realisierte Schutz in Form der Copyleft-Klausel voll durch. Andererseits kann aber die proprietäre Nutzung, der durch ein eigenständiges Patent geschützten Funktionalität, selbst nicht verhindert werden, denn insoweit geht die Wirkung der Copyleft-Klausel ins Leere. Anders ist die Situation freilich dann zu beurteilen, wenn es sich bei dem angemeldeten Patent ausnahmsweise um ein sog. abhängiges Patent handelt. Ein abhängiges Patent liegt vor, wenn dieses wesentliche Erfindungsmerkmale eines älteren Patents benutzt und ohne diese Benutzung nicht ausgeführt werden kann.<sup>279</sup> Zwar berührt die Abhängigkeit eines Patents dessen Erteilungsfähigkeit nicht, wohl aber ist die Ausübung eines solchen Patents von der Zustimmung des Inhabers des älteren Patents abhängig.<sup>280</sup> Wurde also bereits ein Patent auf ein Open Source Programm erteilt und ist das neue Patent

---

<sup>277</sup> Siehe oben IV.3.

<sup>278</sup> Nach *Schiffner*, OSS, S. 162ff., ergibt sich der Copyleft-Effekt aus einer inhaltlichen Beschränkung des Nutzungsrechts.

<sup>279</sup> *Benkhard-Bruchhausen*, PatG, § 9, Rn. 72.

<sup>280</sup> BGH GRUR 1964, 606 (611); *Benkhard-Bruchhausen*, PatG, § 9, Rn. 77.

von diesem abhängig, so kann der Patentinhaber des älteren Patents seine Zustimmung zur Nutzung des neuen Patents davon abhängig machen, dass dieses ebenfalls unter den Bedingungen der entsprechenden Copyleft-Lizenz zur Nutzung freigegeben wird. Ihrem Wortlaut nach erfasst § 2 b) GPL, insbesondere unter Rückgriff auf die Präambel, auch die Lizenzierung von Patenten.<sup>281</sup> Hält sich der Patentinhaber des neuen Patents nicht an die Vorgaben des § 2 b) GPL, so ist ihm die Benutzung des älteren Patents und damit mittelbar auch die Nutzung des neueren Patents gemäß § 4 S.2 GPL untersagt.

Wie sich die duale Lizenzierung eines, auf der Weiterentwicklung eines Copyleft geschützten Open Source Programms beruhenden Patents auf freiwilliger Basis realisieren lässt, zeigt die „Open RTLinux Patent License“.<sup>282</sup> Die Lizenz gestattet die lizenzgebührenfreie Benutzung des RTLinux Patents für sämtliche Software, welche unter der GPL lizenziert ist. In der Präambel der „Open RTLinux License“ findet sich aber der Hinweis, dass für sämtliche Software, die nicht unter der GPL lizenziert wird, eine andere (proprietäre) Lizenz erforderlich ist. Dies stellt nach obigen Ausführungen eine zulässige, weil nicht mehr von der Wirkung der Copyleft-Klausel erfasste proprietäre Nutzung des Patents dar.

#### **d. Ergebnis**

Softwarepatente stellen durchaus ein Risiko für die Entwicklung und den Vertrieb von Open Source Software dar. Allerdings ist die Open Source Bewegung den Gefahren durch Softwarepatente keineswegs schutzlos ausgeliefert. Unberechtigte Patentanmeldungen auf proprietäre Software lassen sich durch öffentlich zugängliche Dokumentationssysteme wirksam bekämpfen. Hier sollten weitere Anstrengungen unternommen werden, um die Recherchierbarkeit solcher Systeme zu erhöhen. Zudem sollte auf eine möglichst umfassende Einbeziehung der Patentämter in den Dokumentationsprozess hingearbeitet werden. Patentanmeldungen aus den eigenen Reihen zur proprietären Nutzung können zwar nicht unterbunden werden, keinesfalls kann dadurch aber ein freies

---

<sup>281</sup> So auch *Jaeger/Metzger*, OSS, S. 124.

<sup>282</sup> Open RTLinux Patent License 2.0 vom 05.10.2001 abrufbar unter:  
[http://www.fsmlabs.com/products/rtlinuxpro/rtlinux\\_patent.html](http://www.fsmlabs.com/products/rtlinuxpro/rtlinux_patent.html).

Programm in ein proprietäres umgewandelt werden. Wie das Beispiel RTLinux zeigt, stellt die duale Lizenzierung eines Patents sowohl für die freie als auch für proprietäre Nutzung durchaus einen gangbaren Weg dar. Hierbei kommt es letztlich auf die Bereitschaft des jeweiligen Patentinhabers an, inwiefern er einer derartigen Lösung zustimmen wird. Die Gefahr bestehende proprietäre Patente zu verletzen ist real und wird sich mit jedem erteilten Patent erhöhen. Zwar drohen Schadensersatzansprüche lediglich großen Unternehmen, die sich auf die Entwicklung und den Vertrieb von Open Source Software spezialisiert haben. Unterlassungsansprüche hingegen können gegen alle Entwickler von Open Source Software geltend gemacht werden. Die gesetzlichen Ausnahmetatbestände des § 11 Nr.1 und Nr. 2 PatG haben nur geringe praktische Bedeutung. Die Möglichkeit bestehende proprietäre Patente zu verletzen, kann daher zu einer empfindlichen Behinderung der künftigen Entwicklung von Open Source Software führen. An dieser Stelle zeigt sich besonders deutlich, dass zwischen dem Open Source Modell und der proprietären Softwareentwicklung hinsichtlich des Schutzes gegen bestehende Patente ein strukturelles Ungleichgewicht besteht. Während proprietäre Softwareunternehmen in aller Regel in der Lage sind, sich durch eine mehr oder minder aktive Patentrechtspolitik, insbesondere durch aufwendige Recherchen und den Aufbau eines eigenen Patentfolios und entsprechende Kreuz-Lizenzierungen, gegen die Risiken von Patentverletzungen schützen können, sind Open Source Entwickler in dieser Hinsicht klar benachteiligt.<sup>283</sup> Im Folgenden soll daher untersucht werden, ob und inwiefern dieses strukturelle Ungleichgewicht durch geeignete Maßnahmen seitens der Open Source Bewegung und des Gesetzgebers verringert werden kann.

### **3. Lösungsvorschläge zur Verringerung des strukturellen Ungleichgewichts zwischen dem Open Source Modell und der proprietären Softwareentwicklung**

Es hat sich gezeigt, dass hinsichtlich den Gefahren durch Softwarepatente zwischen dem Open Source Software Modell und der proprietären Softwareentwicklung ein strukturelles Ungleichgewicht

---

<sup>283</sup> Lutterbeck/Gehring/Horns, Kurzgutachten, S. 102; Morgan, Chaining OSS, S. 4; Working Group on Libre Software: „Free Software/Open Source: Information Society Opportunities for Europe?“, April 2000, Version 1.2, S. 21; abrufbar unter: <http://eu.conecta.it/paper.pdf>

besteht. In diesem Abschnitt sollen verschiedene Lösungsvorschläge zur Verringerung dieses Ungleichgewichts vorgestellt und untersucht werden. Dabei werden zunächst solche Vorschläge aufgezeigt, die sich auf Softwarepatente im Allgemeinen beziehen. Danach folgt eine Darstellung von Open Source Software spezifischen Lösungsvorschlägen.

#### **a. Softwarepatente im Allgemeinen**

##### **aa. Verbesserung der Prüfungsverfahren und Recherchemöglichkeiten**

Wie bereits mehrfach angesprochen wurde, bestehen bei computerbezogenen Erfindungen nach wie vor erhebliche Defizite bei der Prüfung der Neuheit und Erfindungshöhe einer angemeldeten Erfindung.<sup>284</sup> Dies hängt zum einen mit den mangelnden Recherchemöglichkeiten nach dem Stand der Technik von computerbezogenen Erfindungen zusammen, zum anderen aber auch mit einer ungenügenden Fachausbildung der Patentprüfer. Da sich die Patentierung von computerbezogenen Erfindungen zunächst sehr zögerlich entwickelte, fehlt es noch an einschlägiger Patentfachliteratur. Größtenteils sind die Patentämter daher gegenwärtig auf sonstige Offenbarungen angewiesen, die sich wesentlich schwieriger recherchieren lassen. Als Lösung bietet sich hier eine frühe und rege Beteiligung der Patentämter an öffentlichen Dokumentationsstellen für computerbezogene Erfindungen an. Das Internet bietet hier eine kostengünstige und weltumspannende Informationsquelle. Durch ihre Beteiligung an Dokumentationsstellen können die Patentämter von vorn herein auf die Darbietung der gespeicherten Informationen Einfluss nehmen, um so eine effektivere Recherche zu garantieren. Wichtig ist auch eine verbesserte Schulung des Personals der Patentämter, sowie die Einstellung von spezialisierten Fachkräften. Diese Maßnahmen sollten eine effektivere und schnellere Patentprüfung ermöglichen und die Gefahr der Erteilung von Trivialpatenten möglichst eindämmen.

---

<sup>284</sup> *Esslinger/Betten*, CR 2000, 18 (21); *Heide*, CR 2003, 165 (170f.); *Horns*, GRUR 2001, 1 (13); *Marly*, Computersoftware in der EU, S. 52; *Nack/Straus*, MPI-Gutachten, S. 207f.; *Winischhofer*, Computersoftware und Patentrecht, S. 101ff.

## **bb. Verkürzung der Laufzeit von Softwarepatenten**

Nach der geltenden Rechtslage beträgt die Laufzeit eines Patents gemäß § 16 I PatG zwanzig Jahre, beginnend mit dem Tag, der auf die Anmeldung der Erfindung folgt. Bedenkt man, dass vor ungefähr zwanzig Jahren gerade der IBM PC mit Intels 8088 CPU auf den Markt kam, dann erschließt sich selbst dem Laien, dass Innovationen im Bereich der Softwareindustrie in wesentlich schnelleren Abständen erfolgen. Es wurde daher bereits mehrfach vorgeschlagen, die Laufzeit eines Softwarepatents durch eine Verkürzung auf 2 bis 5 Jahre den Gegebenheiten der Softwareindustrie anzupassen.<sup>285</sup> Zu bedenken ist allerdings, dass die Laufzeit eines Patents nicht erst ab der Erteilung, sondern bereits ab der Anmeldung zu laufen beginnt. Da allein schon der Patenterteilungsprozess mehrere Jahre dauern kann, wäre bei einer kürzeren Laufzeit, das Patent häufig schon vor der Erteilung abgelaufen. Bedeutsamer ist jedoch die Tatsache, dass eine Verkürzung der Patentlaufzeit nur für computerbezogene Erfindungen mit den Vorgaben des TRIPS Übereinkommens nicht vereinbar wäre. Gemäß Art. 33 TRIPS ist für alle Erfindungen eine Patentlaufzeit von zwanzig Jahren vorzusehen. Insgesamt stellt daher einer Verkürzung der Patentlaufzeit keine geeignete Maßnahme zur Bewältigung der Gefahren von Softwarepatenten dar.<sup>286</sup>

## **b. Open Source Software spezifische Lösungsvorschläge**

Bevor auf die Frage einzugehen ist, welche Möglichkeiten es auf legislatorischer Ebene gibt, um Open Source Software gegen die Auswirkungen von Softwarepatenten zu schützen, stellt sich die Frage, ob dieses Entwicklungsmodell aus Sicht des Gesetzgebers überhaupt schutz- und förderungswürdig ist. Neben den mit dem Einsatz von Open Source Software verbundenen Kostenreduzierungen, sprechen vor allem Aspekte der Systemstabilität und Systemsicherheit für dieses Entwicklungsmodell. Es kann mittlerweile als gesichertes Erkenntnis angesehen werden, dass Open Source Software sicherer und stabiler ist als vergleichbare proprietäre Software.<sup>287</sup> Aufgrund seiner offenen Struktur und seiner

---

<sup>285</sup> So z.B. *Jeff Bezos*, Gründer von Amazon.com, vgl. [http://www.oreilly.com/news/amazon\\_patents.html](http://www.oreilly.com/news/amazon_patents.html).

<sup>286</sup> So auch *Lutterbeck/Horns/Gehring*, Kurzgutachten, S. 132.

<sup>287</sup> *Gehring*, IT-Security, S. 16ff.; *Grassmuck*, Freie Software, S. 361ff.; *Horns*, JurPC Web-Dok. 223/2000, Abs. 58; *Omsels*, FS-Hertin, S. 143; *Sandl*, CR 2001, 346 (348f.).

leichten Zugänglichkeit können Fehler und Unzulänglichkeiten im Code viel schneller erkannt und beseitigt werden als bei proprietärer Software. Der offene Quellcode ermöglicht es Experten auch, Software, welche für den Einsatz in sicherheitssensitiven Bereichen vorgesehen ist, nach Sicherheitsmängeln oder böartigem Code zu untersuchen.<sup>288</sup> Es sprechen somit handfeste Argumente für den Schutz und die Förderung von Open Source Software. Hinzu kommt, dass sich gerade Deutschland, was die Entwicklung und den Einsatz von Open Source Software anbelangt, an die Weltspitze manövriert hat.<sup>289</sup>

#### **aa. Erweiterung der Ausnahmetatbestände des § 11 PatG**

Zunächst ließe sich zum Schutz von Open Source Software an eine Erweiterung der Ausnahmetatbestände des § 11 PatG denken.

#### **(1) Privilegierung der nichtgewerblichen Nutzung außerhalb des privaten Bereichs**

*Nack/Straus* haben eine Erweiterung des § 11 Nr. 1 PatG vorgeschlagen, nach der auch die nichtgewerbliche Nutzung von Open Source Software außerhalb des privaten Bereichs privilegiert werden soll.<sup>290</sup> Die Tragweite einer solchen Privilegierung dürfte jedoch weniger weit reichen, als die Autoren es vermuten mögen. So würden m.E. wichtige Bereiche der Nutzung von Open Source Software von diesem Privileg überhaupt nicht erfasst werden, wie z.B. die Nutzung in „Embedded Systems“ oder der kommerzielle Vertrieb von Open Source Distributionen. In beiden Fällen dürfte nämlich ohne weiteres ein gewerbliches Handeln anzunehmen sein. Ein gewerbliches Handeln setzt generell voraus, dass am Markt planmäßig und dauerhaft Waren oder Leistungen gegen Entgelt angeboten werden.<sup>291</sup> Auf eine Gewinnerzielungsabsicht kommt es dabei nicht unbedingt an.<sup>292</sup> Auch wenn für die Nutzung eines Open Source Programms keine Lizenzgebühren erhoben werden, so folgt daraus nicht zwangsläufig das Vorliegen eines nichtgewerblichen Handelns. Unternehmen wie SuSe, RedHat oder Mandrake verfolgen sehr wohl wirtschaftliche Zwecke und sind demzufolge als

---

<sup>288</sup> *Horns*, JurPC Web-Dok. 223/2000, Abs. 59; *Sandl*, CR 2001, 346 (349).

<sup>289</sup> *Lutterbeck/Gehring/Horns*, Kurzgutachten, S. 3; *Sandl*, CR 2001, 346 (347).

<sup>290</sup> *Nack/Straus*, MPI-Gutachten, S. 228.

<sup>291</sup> *Palandt-Heinrichs*, BGB, § 14, Rn. 2.

<sup>292</sup> *Palandt-Heinrichs*, BGB, § 14, Rn. 2.

gewerbliche Anbieter anzusehen. Da somit entscheidende Bereiche der Nutzung von Open Source Software überhaupt nicht erfasst werden, erscheint eine derartige Privilegierung wenig sinnvoll. Daneben bestehen – das räumen auch *Nack/Straus* ein – Probleme in Bezug auf die Vorgaben des TRIPS Übereinkommens. So ermöglicht Art. 30 TRIPS zwar begrenzte Ausnahmen von den Wirkungen eines Patents, sofern solche Ausnahmen nicht unangemessen im Widerspruch zur normalen Verwertung des Patents stehen und die berechtigten Interessen des Inhabers des Patents nicht unangemessen beeinträchtigen. Es ist aber zweifelhaft, ob die von *Nack/Straus* vorgesehene Erweiterung des § 11 Nr. 1 PatG mit Art. 30 TRIPS vereinbar ist, insbesondere im Hinblick auf Art. 27 Nr. 1 S.2 TRIPS, wonach bei der Ausübung eines Patents nicht nach dem Gebiet der Technik diskriminiert werden darf.

## **(2) Quelltextprivileg**

Denkbar wäre auch die Einführung eines Quelltextprivilegs mit dem Inhalt, dass das Herstellen, Anbieten, In-Verkehr-bringen, Besitzen oder Einführen von Software im Quellcode vom Patentschutz ausgenommen wird.<sup>293</sup> Die gewerbliche Nutzung des Programms durch Ausführung auf einem Computer wäre jedoch nach wie vor vom Patent erfasst. Dadurch wäre die Entwicklung und der Vertrieb von Open Source Software gänzlich von der Wirkung eventuell bestehender Patente ausgenommen. Andererseits würde dadurch die Rechtsverfolgung für Patentinhaber erheblich erschwert werden, da sie nunmehr einzeln gegen, mitunter schwer zu ermittelnde gewerbliche Nutzer der Software vorgehen müssten. Im Ergebnis bestehen daher auch gegen ein Quelltextprivileg Bedenken im Hinblick auf die Vorgaben des TRIPS Übereinkommens. Eine entsprechende Änderung ließe sich realistischere Weise nur auf internationaler Ebene realisieren.

## **bb. Erteilung von Zwangslizenzen gemäß § 24 I PatG**

Zu denken wäre auch – quasi als „ultimo ratio“ – an die Vergabe von Zwangslizenzen im Sinne des § 24 PatG.<sup>294</sup> Voraussetzung für die Erteilung einer Zwangslizenz gemäß § 24 I PatG ist, dass der

---

<sup>293</sup> *Horns*, JurPC Web-Dok. 223/2000, Abs. 75f.; *Lutterbeck/Horns/Gehring*, Kurzgutachten, S. 132f.

<sup>294</sup> So auch *Nack/Straus*, MPI-Gutachten, S. 231.

Lizenzsucher sich innerhalb eines angemessenen Zeitraumes erfolglos bemüht hat, vom Patentinhaber die Zustimmung zu erhalten, die Erfindung zu angemessenen geschäftsüblichen Bedingungen zu benutzen (Nr. 1) und das öffentliche Interesse die Erteilung einer Zwangslizenz gebietet (Nr. 2). Aus dem Wortlaut des § 24 PatG ergibt sich bereits, dass die Erteilung einer Zwangslizenz nur dann in Betracht kommt, wenn seitens eines Open Source Entwicklers ein angemessenes Angebot zur Lizenzierung eines proprietären Softwarepatents vorgelegen hat und dieses abgelehnt wurde. Daran dürfte es jedoch regelmäßig fehlen.<sup>295</sup> Liegt dieses Tatbestandsmerkmal ausnahmsweise vor, erfordert § 24 PatG darüber hinaus noch das Vorliegen eines öffentlichen Interesses, welches die Erteilung einer Zwangslizenz gebietet. Bei der Ermittlung eines öffentlichen Interesses bedarf es einer sorgfältigen Abwägung der betroffenen Interessen im Einzelfall, wobei vor allem technische, wirtschaftliche und sozialpolitische Gesichtspunkte herangezogen werden können.<sup>296</sup> Hier kann durchaus im Wege einer gesamtwirtschaftlichen Betrachtung zugunsten einer Open Source Entwicklung entschieden werden.<sup>297</sup> Insgesamt erscheint die Erteilung von Zwangslizenzen jedoch nur ausnahmsweise geeignet, das strukturelle Ungleichgewicht zwischen dem Open Source Modell und proprietäre Software zu verringern.

### **cc. Open Patents Pool und Einführung einer Neuheitsschonfrist**

Die bisherigen Lösungsvorschläge bezogen sich durchweg auf eine Einschränkung des Schutzbereichs von Softwarepatenten zugunsten von Open Source Entwicklungen. Es fragt sich, ob die Entwickler von Open Source Software das Patentsystem nicht auch zu ihren Zwecken nutzbar machen können. Auf den ersten Blick mag ein solcher Vorschlag angesichts der prinzipiell völlig gegenläufigen Zielrichtungen des Open Source Modells und des protektionistischen Patentrechts absurd erscheinen. So werden denn auch Softwarepatente und Open Source Software als zwei schlicht inkompatible Konzepte bezeichnet.<sup>298</sup> Zu bedenken ist allerdings,

---

<sup>295</sup> So verlangt *Thompson Multimedia* allein für die Lizenzierung eines mp3-decoders pauschal eine Lizenzgebühr von 50.000 USD, vgl. <http://www.mp3licensing.com/royalty/index.html>.

<sup>296</sup> *Benkhard-Rogge*, PatG, § 24, Rn. 16ff.

<sup>297</sup> Vgl. auch *Horns*, JurPC Web-Dok. 223/2000, Abs. 78.

<sup>298</sup> *Riek*, Softwarepatente gefährden Fortschritt und Wettbewerb; <http://www.sicherheit-im-internet.de/themes/print.phtml?ttid=2&tid=74>.

dass auch das Urheberrecht grundsätzlich eine andere Zielrichtung verfolgt als das Open Source Modell. Dennoch wäre das gesamte Open Source Modell ohne Rückgriff auf das Urheberrecht überhaupt nicht realisierbar. Es ist also gelungen das Urheberrecht für die Verwirklichung des Open Source Modells zu instrumentalisieren. Natürlich lässt sich das Urheberrecht nicht direkt mit dem Patentrecht vergleichen, da beide Normgefüge zwei völlig unterschiedliche Schutzrichtungen verfolgen. Es fragt sich aber dennoch, ob nicht auch das Patentrecht für die Zwecke der Open Source Bewegung tauglich gemacht werden kann. Zu denken wäre dabei an die Schaffung eines eigenen Open Patents Pools.<sup>299</sup> Derartige „Open Patents“ könnten dann durch eine an die GPL oder andere Copyleft-Lizenzen angelehnte Lizenz<sup>300</sup> kostenfrei an alle anderen Open Source Entwickler lizenziert werden. Jede proprietäre Nutzung eines Patents wäre dann grundsätzlich ausgeschlossen. Darüber hinaus bestünde für einzelne Open Source Projekte auch die Möglichkeit durch „Cross-Licensing“ auch proprietäre Patente für sich nutzen. Ein derartiges Konzept verfolgt z.B. der GNU/Linux Distributor RedHat.<sup>301</sup> Wegen des überwiegend dezentralen Schaffungsprozesses von Open Source Software, wäre es erforderlich eine zentrale Treuhandinstanz<sup>302</sup> zu schaffen, welche sich um die Anmeldung und Durchsetzung, sowie das „Cross-Licensing“ von Patenten kümmert. Ansätze zur Bildung von Treuhandinstanzen in der Open Source Gemeinde sind bereits jetzt zu beobachten. So bietet etwa die FSF Europe eine sog. Treuhänderische Lizenzvereinbarung an, mittels derer Autoren von Open Source Software ihre Rechte an die FSF Europe übertragen können, damit diese als juristische Interessenvertretung fungieren kann.<sup>303</sup> Wie die teilweise recht kostenintensiven Patentaktivitäten einer solchen Treuhandinstanz zu finanzieren wären, müsste freilich noch geklärt werden. Angesichts der wirtschaftlich durchaus erfolgreichen GNU/Linux-Distributionen, erscheint es jedoch nicht unwahrscheinlich, dass auch hier ein ertragsstarkes Geschäftsmodell gefunden werden kann. Hat sich erstmal ein gewisses Volumen an

---

<sup>299</sup> Vgl. Cooperative OSS Patent Pool, <http://home.netcom.com/~kmsself/OSSPatPool/index.html>; *Gehring*, Berliner Ansatz zu “Open Software Patents”, 13. Mai 2000, <http://www.sicherheit-im-internet.de/themes/themes.phtml?tid=2&tdid=75>.

<sup>300</sup> Vgl. etwa <http://www.openpatents.org>.

<sup>301</sup> Vgl. [http://www.redhat.com/legal/patent\\_policy.html](http://www.redhat.com/legal/patent_policy.html).

<sup>302</sup> *Gehring*, Berliner Ansatz zu “Open Software Patents”, 13. Mai 2000, <http://www.sicherheit-im-internet.de/themes/themes.phtml?tid=2&tdid=75>.

<sup>303</sup> <http://www.fsfeurope.org/projects/fla/fla.de.html>.

Patenten aufgebaut, so wäre es der Open Source Bewegung möglich durch eine defensive Patentpolitik den Gefahren durch Softwarepatente wirkungsvoll zu begegnen, ohne dabei die Grundlagen des eigenen Entwicklungsmodells zu gefährden.

De lege lata besteht für die Entwickler von Open Source Software zur Zeit aber ein entscheidendes Problem beim Aufbau eines eigenen Patentportfolios. Nach der geltenden Rechtslage in Deutschland führt nämlich auch jede Vorveröffentlichung des Erfinders selbst zum Wegfall der Neuheit einer Erfindung. Im Prinzip ist ein Erfinder damit gezwungen seine Erfindung bis zur Anmeldung geheim zu halten. Eine derartige Vorgehensweise ist jedoch mit dem auf Wissenstransfer aufgebauten Open Source Software Modell nicht zu vereinbaren. Nach der geltenden Rechtslage ist es daher für Open Source Entwickler praktisch unmöglich eigene Patente auf ihre Entwicklungen anzumelden. Hier könnte die Einführung einer Neuheitsschonfrist, wie sie beispielsweise in den USA<sup>304</sup> („sog. grace period“) existiert, Abhilfe schaffen.<sup>305</sup> Zweck einer Neuheitsschonfrist ist es, jede Veröffentlichung des Erfinders selbst, die innerhalb einer bestimmten Frist vor der Patentanmeldung erfolgt ist, als unschädlich zu betrachten. Für einen Konkurrenten, der eine Erfindung zum Patent anmelden will, gilt die Veröffentlichung des Erfinders hingegen als zum Stand der Technik gehörend und kann diesem daher entgegengehalten werden. Die Neuheitsschonfrist schützt also nur den Erfinder, der selbst Teile seiner Erfindung veröffentlicht hat. Durch die Einführung einer solchen Neuheitsschonfrist wäre es Open Source Entwicklern also möglich, ihr bisheriges Entwicklungsmodell beizubehalten, ohne dabei die Möglichkeit einer Patentanmeldung einzubüßen. In einem anderen, durchaus vergleichbaren Kontext wird die Einführung einer Neuheitsschonfrist in Deutschland bzw. in Europa schon seit längerem befürwortet. Nach einer vom Bundesministerium für Bildung und Forschung (BMBF) in Auftrag gegebenen im November 2001 veröffentlichten Studie zur Patentierung von Hochschulforschungsergebnissen wurde die Einführung einer Neuheitsschonfrist uneingeschränkt empfohlen.<sup>306</sup> Im Anschluss an die Studie hat sich BMBF in einem gemeinsamen

---

<sup>304</sup> Title 35 Sektion 102 (b) USC.

<sup>305</sup> Auch in Deutschland existierte bis 1978 eine derartige Neuheitsschonfrist.

<sup>306</sup> Hausberg, BMBF-Studie zur Neuheitsschonfrist, S. 7ff.

Standpunkt<sup>307</sup> mit dem Bundesministerium für Justiz (BMJ), und dem Bundesministerium für Wirtschaft und Technologie (BMWi) für die Einführung einer Neuheitsschonfrist in das europäische Patentrecht ausgesprochen. Die Ergebnisse der Studie lassen sich insoweit auf die Situation der Open Source Bewegung übertragen, als sowohl im Hochschulbereich als auch bei der Entwicklung von Open Source Software ein breiter Wissenstransfer und eine frühzeitige Veröffentlichung von essentieller Wichtigkeit sind und von daher ähnliche Schwierigkeiten in Bezug auf die Anmeldung von Patenten bestehen. Zur Verringerung des strukturellen Ungleichgewichts zwischen dem Open Source Software Modell und des proprietären Softwareentwicklungsmodells betreffend Softwarepatente sollte daher die Einführung einer Neuheitsschonfrist möglichst auf europäischer Ebene angestrebt werden.<sup>308</sup> Nach Einführung einer solchen Neuheitsschonfrist hätten prinzipiell beide Entwicklungsmodelle gleichermaßen die Möglichkeit Softwarepatente zu ihren Zwecken einzusetzen. Sind die Grundlagen für eine derartige defensive Patentstrategie der Open Source Bewegung erstmal geschaffen, dürfte es den Entwicklern von Open Source Programmen auch nicht mehr schwer fallen, ihre grundlegend ablehnende Haltung gegenüber Softwarepatenten zu überwinden, um diese für sich nutzbar zu machen.

## **VI. Fazit und Ausblick**

Trotz anders lautender Befürchtungen, werden Patente für computerbezogene Erfindungen nicht zum Untergang der Open Source Bewegung führen. Zwar bergen Softwarepatente durchaus Risiken für die Entwicklung von Software allgemein und von Open Source Software im Besonderen, diese Gefahren sollten jedoch nicht überzeichnet werden.<sup>309</sup> Nach der geltenden Rechtslage und Erteilungspraxis in Deutschland ist nicht zu erwarten, dass es zu einer inflationären Erteilung von Trivialpatenten kommt, welche Innovationen und Wettbewerb im Bereich der Softwareentwicklung zum Erliegen bringen wird. Im Gegensatz zu den USA, wo jede computerbezogene Erfindung, die einen nützlichen, konkreten und

---

<sup>307</sup> [http://www.patente.bmbf.de/de/patentpo\\_1174.php](http://www.patente.bmbf.de/de/patentpo_1174.php).

<sup>308</sup> *Lutterbeck/Horns/Gehring*, Kurzgutachten, S. 133, schlagen eine Neuheitsschonfrist von 12 Monaten Dauer vor.

<sup>309</sup> *Esslinger/Betten*, CR 2000, 18 (21); *Jaeger/Metzger*, OSS, S. 128; *Schiffner*, OSS, S. 53.

greifbaren Erfolg („useful, concrete and tangible result“)<sup>310</sup> aufweist, als patentierbar gilt, bestehen nach der geltenden Rechtslage in Deutschland durch das Kriterium der Technizität hohe Anforderungen an die Patentierbarkeit von Software. Die in der Entscheidung „*Sprachanalyseeinrichtung*“ nahezu uneingeschränkte Ausweitung der Patentierfähigkeit computerbezogener Erfindungen wurde vom BGH in der Entscheidung „*Suche fehlerhafter Zeichenketten*“ korrigiert und erheblich eingeschränkt. Wie die neuere Rechtsprechung des BPatG zeigt, wurde diese restriktivere Linie bisher auch konsequent umgesetzt. Nach dem gegenwärtigen Stand der Dinge ist auch nicht zu erwarten, dass sich an dieser Situation nach einer Umsetzung der EU-Richtlinie zur Patentierbarkeit von computer-implementierten Erfindungen entscheidend etwas ändern könnte. Die aktuelle Fassung des Richtlinienvorschlags führt ihrem Ziel und Inhalt nach lediglich zu einer Harmonisierung der geltenden europäischen Erteilungspraxis.

Aufgrund der unterschiedlichen Entwicklungsmodelle von Open Source Software und proprietärer Software, ist zur Zeit allerdings ein strukturelles Ungleichgewicht zu Lasten des Open Source Software Modells zu konstatieren. Sollte es nicht gelingen dieses Ungleichgewicht zu verringern, so ist davon auszugehen, dass künftige Entwicklungen der Open Source Gemeinde mitunter empfindlich durch bestehende proprietäre Softwarepatente behindert werden könnten. Einer solchen Entwicklung sollte angesichts der offensichtlichen Stärken und Vorteilen des Open Source Software Modells und der überragenden Stellung Deutschlands im Bereich der Entwicklung und Nutzung von Open Source Software nicht tatenlos zugesehen werden. Legislatorschen Maßnahmen zum Schutz von Open Source Software seitens des Gesetzgebers sind aufgrund internationaler Verpflichtungen jedoch enge Grenzen gesetzt. Es hat sich jedoch gezeigt, dass die Open Source Gemeinde durch die Schaffung eines eigenen Patentportfolios in der Lage wäre, den Gefahren durch Softwarepatente im Wege einer defensiven Patentpolitik wirksam zu begegnen. Flankiert werden müssten derartige Anstrengungen jedoch durch die (Wieder-) Einführung einer Neuheitsschonfrist für Patentanmeldungen auf deutscher bzw.

---

<sup>310</sup> US Court of Appeals for the Federal Circuit, Urt. v. 23.07.1998 - „*State Street Bank & Trust v. Signature Financial Group*“; vgl. dazu *Esslinger/Hössle*, Mitt. 1999, 327ff.

europäischer Ebene. Nur so kann das auf breitem Wissenstransfer und früher Veröffentlichung aufgebaute und dadurch so erfolgreiche Open Source Software Modell das Patentrecht auch für seine Zwecke in Anspruch nehmen. Daneben sollte die Qualität des Prüfungsverfahrens der Patentämter deutlich verbessert werden, umso den besonderen Anforderungen in Bezug auf die Feststellung der Neuheit und erfinderischen Tätigkeit einer computerbezogenen Erfindung gerecht zu werden. Entscheidend ist hierbei insbesondere eine Verbesserung der Recherchemöglichkeiten der Patentämter. Durch eine konsequente Umsetzung dieser Maßnahmen sollte es gelingen, die Gefahren durch Softwarepatente für die Softwareindustrie im Ganzen und der Open Source Gemeinde im Speziellen auf ein vertretbares Maß zu reduzieren und so den technischen Fortschritt auf dem Gebiet der computerbezogenen Erfindungen zum Wohle der Gesellschaft als Ganzes zu fördern.