ECONSTOR
*Make Your Publications Visible.*

A Service of

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Maaß, Christian; Scherm, Ewald

**Article — Published Version**

# Die Ablehnung der Softwarepatentrichtlinie: Ein zweifelhafter Erfolg

Wirtschaftsdienst

*Suggested Citation:* Maaß, Christian; Scherm, Ewald (2005) : Die Ablehnung der Softwarepatentrichtlinie: Ein zweifelhafter Erfolg, Wirtschaftsdienst, ISSN 0043-6275, Springer, Heidelberg, Vol. 85, Iss. 10, pp. 660-663, https://doi.org/10.1007/s10273-005-0419-2

This Version is available at:
https://hdl.handle.net/10419/42412

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

DOI: 10.1007/s10273-005-0419-2



Christian Maaß, Ewald Scherm

# Die Ablehnung der Softwarepatentrichtlinie: Ein zweifelhafter Erfolg!

*Anfang Juli 2005 hat das Europäische Parlament die Softwarepatentrichtlinie zurückgewiesen. Nach welchen Regeln werden computerimplementierte Erfindungen derzeit behandelt? Welche Folgen hat die Ablehnung der Richtlinie für die zukünftige Behandlung von Software-Entwicklungen?*

Die Softwarelizenzierung beruht in Europa auf dem Urheberrecht. Danach genießt nur eine bestimmte Programmimplementierung – nicht aber das Verfahren bzw. die Idee als solche – urheberrechtlichen Schutz. Bei einem Patent ist es dagegen möglich, die hinter der Software stehende Idee zu schützen, wodurch der Patentinhaber ein gegenüber Dritten wirksames Ausschließlichkeitsrecht erhält. Aus volkswirtschaftlicher Sicht ist der Grundgedanke einer solchen Schutzmöglichkeit darin zu sehen, getätigte Investitionen in Forschung und Entwicklung zu schützen und damit den Innovationsanreiz auf Unternehmensseite zu erhöhen. Ferner erfüllen Patente eine Informationsfunktion, da durch die Anmeldung eines Patents die Transparenz im Forschungssektor erhöht wird.

In Europa wird die Patentierung von Software durch die Europäische Patentübereinkunft (EPÜ) geregelt, die grundsätzlich ein Patentierungsverbot von Software „als solche" vorsieht (EPÜ Art. 52, Abs. 2). Es hat sich aber als problematisch erwiesen, dass der Begriff „als solche" nicht definiert ist. So hat z.B. das Europäische Patentamt in den vergangenen Jahren mehr als 30 000 Softwarepatente erteilt, indem es Software als „wesensbestimmenden Bestandteil" eines technischen Geräts interpretierte[1]. Erschwerend kommt hinzu, dass europäische Patente nach Ablauf der Einspruchsfrist nur noch vor nationalen Gerichten für das betreffende Land angefochten werden können. Angesichts der uneinheitlichen Rechtsprechung und Verwaltungspraxis der nationalen Patentämter wurde von der Europäischen Kommission daher im Jahr 2002 ein Richtlinienentwurf bezüglich der Patentierbarkeit computerimplementierter Erfindungen vorgelegt[2]. Dieser zielte darauf, die Gesetze der Mitgliedstaaten zu harmonisieren und sich bei der Patentvergabe an der Praxis des europäischen Patentamtes zu orientieren.

Die Gegner von Softwarepatenten befürchteten durch diese Richtlinie allerdings eine Beeinträchtigung des Wettbewerbs auf dem europäischen Softwaremarkt. Nach ihrer Ansicht liegt das Problem bei der Patentierung von Software darin, dass diese in der Regel sequentiell entwickelt wird und somit auf vorangegangenen Softwarelösungen aufbaut; sind diese patentgeschützt, fallen faktisch zwangsläufig Lizenzgebühren an[3]. Gerade für Entwickler quelloffener Software sowie kleine und mittelständische Unternehmen seien diese aber nur schwer zu finanzieren[4]. Folglich würde die Patentierbarkeit von Software zu erheblichen Wettbewerbsverzerrungen zu Gunsten weniger Großunternehmen führen und sich negativ auf Investitionen in neue Forschungsprojekte auswirken. Demgegenüber argumentierten die Befürworter von Softwarepatenten, dass ein umfassender Patentschutz notwendig sei, da Erfindungen im Softwarebereich Investitionen erfordern und es diese zu schützen gelte .

Am 6. Juli 2005 hat das Europäische Parlament in zweiter Lesung die seit Jahren äußerst umstrittene Softwarepatentrichtlinie zurückgewiesen. Aufgrund der im Vorfeld dargelegten Kritikpunkte verwundert es nicht, dass dies seitens der Patentgegner als Erfolg gefeiert wird: Diese Entscheidung sei ein eindeutiges Votum gegen die Patentierung von Software. Mitunter ist auch von der „Beendigung eines Alptraums" die Rede[5]. Bei genauerer Betrachtung erscheinen

*Prof. Dr. Ewald Scherm, 45, ist Inhaber des Lehrstuhls für Betriebswirtschaftslehre, insbesondere Organisation und Planung, der FernUniversität in Hagen; Christian Maaß, 29, Dipl.-Kaufmann, ist dort wissenschaftlicher Mitarbeiter.*

[1] Vgl. C. Maaß, E. Scherm: Softwarepatente, in: Das Wirtschaftsstudium, 33. Jg. (2004), Nr. 10, S. 1192.

[2] Vgl. Europäische Kommission: Richtlinie des europäischen Parlaments und des Rates über die Patentierbarkeit computerimplementierter Erfindungen, Brüssel 2002.

[3] Vgl. D. Evans, A. Layne-Farrar: Software patents and open source: The battle over intellectual property rights, National Economic Research Associates, Brüssel 2004.

[4] Vgl. G. Jakob: Positionspapier des Vereins zur Förderung Freier Software zum gegenwärtigen Stand der Diskussion um Logik- und Ideenpatente, Lustenau 2003.

derartige Aussagen aber zu pauschal und bei dem vermeintlichen Erfolg handelt es sich eher um einen Pyrrhussieg. Um das zu verdeutlichen, wird im Folgenden die Praxis der Patentvergabe in der Bundesrepublik Deutschland beleuchtet und darauf aufbauend die Ablehnung der Softwarepatentrichtlinie diskutiert.

### Patentierung computerimplementierter Erfindungen in Deutschland

Mit der Ablehnung der Softwarepatentrichtlinie orientiert sich die Patentierung computerimplementierter Erfindungen auch in Zukunft an den Gesetzen der jeweiligen EU-Staaten und der Vorgehensweise des Europäischen Patentamts. Die deutsche Rechtsprechung geht hinsichtlich der Kriterien der Patentfähigkeit einer Erfindung in allen zentralen Punkten mit der europäischen Patentübereinkunft konform. Zur Prüfung des Patentanspruchs werden drei zentrale Kriterien herangezogen[6]:

- Das Erfordernis der *Neuheit* besagt, dass für eine Erfindung nur dann ein Patent erteilt wird, wenn diese gemäß Art. 52, Abs. 1 EPÜ nicht zum Stand der Technik gehört. Das bedeutet, dass die betreffende Erfindung vor ihrem Anmeldedatum beim Patentamt der Öffentlichkeit noch nicht durch schriftliche oder mündliche Beschreibung, durch Benutzung oder in sonstiger Weise zugänglich gemacht worden ist.
- Weiterhin muss eine *erfinderische Tätigkeit* vorliegen (vgl. Art. 56 EPÜ). Damit ist gemeint, dass die Erfindung für einen Fachmann nicht in nahe liegender Weise aus dem Stand der Technik abgeleitet werden kann. Zur Beurteilung der erfinderischen Tätigkeit muss aus der Beschreibung des beantragten Patents hervorgehen, welcher Aufgabe sich der Erfinder gewidmet und wie er diese gelöst hat.
- Schließlich muss eine Erfindung *gewerblich anwendbar*, herstellbar oder benutzbar sein (vgl. Art. 57 EPÜ). Dabei kommt es aber nicht darauf an, dass die Erfindung tatsächlich eingesetzt wird. Für die Erteilung des Patents ist die Angabe hinreichend, in welchem Bereich die praktische Anwendung möglich wäre. Zudem ist es zulässig, dass bis zur Realisierung der Erfindung noch weitere Forschungsaktivitäten anfallen.

Eine Vergabe von Patenten auf computerimplementierte Erfindungen ist angesichts dieser Kriterien allerdings problembehaftet: Sie stellen nicht auf die Frage ab, ob es sich dabei um Software „als solche" handelt, sondern auf den technischen Charakter einer Erfindung. Patentfähig ist die betreffende Software demnach dann, wenn sie zur Lösung eines technischen Problems beiträgt, um – in juristischer Terminologie formuliert – die Naturkräfte beherrschbar zu gestalten[7]. Die verschiedenen Merkmale einer Erfindung werden dabei durch das Europäische Patentamt in ihrer Gesamtheit in Bezug auf den Hauptanspruch bewertet. Anders formuliert kann nach gültiger Rechtsprechung ein Patentschutz auf Software beantragt werden, wenn der Hauptanspruch technischer Natur ist. Vor diesem Hintergrund lässt sich seit geraumer Zeit eine Ausweitung der Patentierbarkeit computerimplementierter Erfindungen beobachten, wobei – in Anlehnung an die Rechtsprechung des Bundesgerichtshofs – zwischen drei Fallgruppen differenziert werden kann[8]:

- In der *ersten Fallgruppe* findet sich Software, die mechanische Einheiten steuert. Derartige Erfindungen erachtet man seit jeher als patentfähig. Ein bekanntes und viel zitiertes Beispiel stellt das Urteil des Bundesgerichtshofs bezüglich der Patentierbarkeit eines Anti-Blockier-Systems dar. Hier ist die Software in einen technischen Gesamtablauf mit mechanischen Einheiten eingebettet und die Messsignale wirken unmittelbar auf die Naturkräfte ein, indem sie das Bremsverhalten des Fahrzeugs beeinflussen.
- Aufgrund der Ganzheitsbetrachtung bei der Patentvergabe existieren aber Abgrenzungsschwierigkeiten bei Softwarelösungen, die nicht in derartiger Weise in einen Gesamtprozess eingebettet sind. Diese sind in der *zweiten Fallgruppe* verortet. Für die Erteilung eines Patents ist es dabei nach gültiger Rechtsprechung hinreichend, dass eine enge Beziehung zwischen dem Programm und den zur Datengewinnung eingesetzten technischen Mitteln besteht. Dies betrifft Programme, die auf die Steuereinheit eines Computers einwirken, wozu im Wesentlichen die Komponenten eines Betriebssystems zählen[9]. Nicht patentfähig ist dagegen Anwendungssoftware, da sie in der Regel nicht in die Steuerung der Hardware eingreift.
- Im Rahmen der *dritten Fallgruppe* zeichnet sich gegenwärtig eine gewisse Ausdehnung der Patentier-

[5] Vgl. hierzu W. Herrmann: Softwarepatente – Ein Alptraum ist vorbei, in: Computerwoche, http://www.computerwoche.de/index.cfm?pageid=254&type=detail&artid=77977.

[6] Vgl. Europäisches Patentamt: Europäisches Patentübereinkommen, München 2002, Art. 52, Abs. 1.

[7] Vgl. G. Benkard, K. Bruchhausen: Patentgesetz, in: G. Benkard, K. Bruchhausen, R. Rogge, A. Schäfers, E. Ullmann (Hrsg.): Patentgesetz, München 1993, S. 137-484, hier § 1 Rn. 44.

[8] Vgl. W. Tauchert: Zur Beurteilung des technischen Charakters von Patentanmeldungen aus dem Bereich der Datenverarbeitung unter Berücksichtigung der bisherigen Rechtsprechung, in: Gewerblicher Rechtschutz und Urheberrecht, 1997, Nr. 2, S. 149-155; T. Jaeger, A. Metzger: Open Source Software, München 2002.

barkeit von Software ab. Der technische Charakter wird durch den Bundesgerichtshof bereits bejaht, wenn technische Überlegungen lediglich vorhanden sind[10]. Das bedeutet faktisch eine Abkehr von der bisherigen Auslegung der Anforderungen, die an die Patentvergabe zu stellen sind. Von der Patentierung sind lediglich solche Softwarelösungen ausgeschlossen, bei denen die Erfindung nicht auf einem technischen Gedanken im Sinne der Naturwissenschaften beruht. „Letztendlich ist nun beliebig interpretierbar, was technische Überlegungen sein sollen."[11] Bei entsprechender Formulierung des Patentantrags ist insofern bereits zum gegenwärtigen Zeitpunkt nahezu jede Softwarelösung patentierbar[12].

**Diskussion**

Da in Europa bereits seit geraumer Zeit Patente auf Software beantragt und erteilt werden, verwundert es, dass der Erhalt des Status quo seitens der Patentgegner als Erfolg gewertet wird: Angeblich behindern Softwarepatente Investitionen in die Forschung und stellen eine Gefahr für die quelloffene Softwareentwicklung sowie für kleine und mittelständische Unternehmen dar. Es liegt daher nahe, vor dem Hintergrund der gegenwärtigen Rechtsprechung zwei zentralen Fragen nachzugehen:

- Wie wirken sich Softwarepatente auf das Investitionsvolumen im Bereich der Forschung und Entwicklung aus?
- Inwieweit behindern Patente die quelloffene Softwareentwicklung und stellen eine Gefahr für kleine und mittelständische Unternehmen dar?

Die Beantwortung der *ersten Frage* gestaltet sich problematisch, da diese auf theoretischer Ebene äußerst kontrovers diskutiert wird. Beispielsweise argumentiert Wiebe, dass es möglicherweise nicht zu einem Monopol im Bereich der Betriebssystemsoftware durch das Unternehmen Microsoft gekommen wäre, wenn kleine Mitbewerber ihre Ideen durch Patente hätten schützen können[13]. Eine derartige Argumentation entbehrt allerdings nicht einer gewissen Willkür, da sie auf der spekulativen Ebene verharrt.

Es liegt vielmehr nahe, den Einfluss von Softwarepatenten auf dem US-amerikanischen Markt zu hinterfragen. Hier ist die Patentierung von Software bereits seit dem Jahr 1981 möglich und es liegen erste empirische Forschungsergebnisse vor. Danach kann bislang keine Innovationswirkung von Softwarepatenten diagnostiziert werden[14]. Zu einem ähnlichen Ergebnis kommt Stolpe, der ferner eine erhöhte Neigung zur Beantragung von gewerblichen Schutzrechten feststellt, wenn Netzeffekte von Bedeutung sind[15]. Bei Gütern, die Netzeffekte aufweisen, hängt der Nutzen der Konsumenten nicht nur vom Preis, sondern auch von dessen Verbreitungsgrad ab. Auf Netzeffektmärkten kommt es deshalb besonders häufig zu einer Monopolbildung. Stolpe vertritt die Auffassung, dass die auf Netzeffektmärkten gegebenen Monopolisierungstendenzen durch gewerbliche Schutzrechte zusätzlich forciert werden.

Diese Ansicht wird auch von der Monopolkommission geteilt, und sie spricht sich dafür aus, keine Ausweitung des Patentschutzes für Software vorzusehen und entsprechenden Tendenzen auf europäischer Ebene eine Absage zu erteilen. „Die mit dem Patentschutz verbundene vorübergehende Monopolstellung eines Unternehmens ist geeignet, die Konzentrationstendenzen auf dem Markt für Softwareprodukte weiter zu verstärken und den Wettbewerb zu behindern."[16] Angesichts dessen verwundert es nicht, dass Patente zunehmend strategisch eingesetzt werden, um auf Basis eines großen Patentportfolios Klagen vorzubeugen[17].

Wenngleich an dieser Stelle aufgrund des frühen Forschungsstadiums in diesem Bereich kein endgültiges Fazit gezogen werden kann, erscheint es mehr als fraglich, dass Softwarepatente Investitionen in die Forschung und Entwicklung forcieren. Dies liegt nicht zuletzt an den kurzen Produktlebenszyklen in dieser Branche; bis zur Erteilung eines Patents vergehen in der Regel mehrere Jahre, während Software häufig

[9] Vgl. J. Betten: Tauschcomputer, in: Gewerblicher Rechtschutz und Urheberrecht, 1992, Nr. 10, S. 600-604; W. Tauchert, a.a.O.

[10] Vgl. J. Betten: Logikverifikation, in: Gewerblicher Rechtschutz und Urheberrecht, 2000, Nr. 6, S. 498-502; J. Betten, Sprachanalyseeinrichtung, in: Gewerblicher Rechtschutz und Urheberrecht, 2000, Nr. 11-12, S. 1007-1010.

[11] Vgl. G. Schölch: Softwarepatente ohne Grenzen, in: Gewerblicher Rechtschutz und Urheberrecht, 2001, Nr. 1, S. 16-21.

[12] Vgl. Gesellschaft für Informatik e. V.: Positionspapier der Gesellschaft für Informatik e. V. zur Patentierung rechnergestützter Erfindungen, Bonn 2005.

[13] Vgl. A. Wiebe: Softwarepatente und Open Source, in: Computer und Recht, 2004, Nr. 12, S. 881-888.

[14] Vgl. J. Bessen, R. Hunt: The software patent experiment, in: OECD conference proceedings: Patents, innovation and economic performance, Paris 2004, S. 247-263.

[15] Vgl. M. Stolpe: Protection against software piracy, in: Economics of Innovation and New Technology, Vol. 9 (2000), S. 25-52.

[16] Vgl. Monopolkommission: Netzwettbewerb durch Regulierung, Baden-Baden 2002, http://www.monopolkom-mission.de/haupt_14/sum_h14_de.pdf.

[17] Vgl. J. Sheehan, C. Martinez, D. Guellec: Understanding business patenting and licensing: Results of a survey, in: OECD conference proceedings: Patents, Innovation and Economic Performance, Paris 2004, S. 89-110.

bereits nach wenigen Monaten als veraltet gilt und durch eine Folgeversion ersetzt wird.

Für die Beantwortung der *zweiten Frage* müssen zunächst die Grundgedanken der quelloffenen Softwareentwicklung skizziert werden. Bei solchen Softwarelösungen – exemplarisch sei das Betriebssystem Linux genannt – werden dem Anwender auf Basis eines Lizenzvertrags sehr umfangreiche Nutzungsrechte zugestanden, die im Gegensatz zum traditionellen Softwaregeschäft stehen. So ist es gestattet, den Quellcode einzusehen, der Aufbau und Arbeitsweise eines Programms offen legt, und Änderungen an diesem vorzunehmen. Die Lizenzbedingungen dieser Software sehen weiterhin vor, dass die Verbreitung quelloffener Software nicht durch Forderungen aus Softwarelizenzen oder Ansprüchen aus Patenten behindert werden darf. Man vertritt die Auffassung, dass die mit der Patentierung von Software einhergehenden Kosten gerade für Entwickler quelloffener Software – die sich teilweise aus ideologischen Gründen und ehrenamtlich in der Softwareentwicklung engagieren – nur schwer zu finanzieren sind[18]. Allerdings darf in diesem Zusammenhang nicht übersehen werden, dass zunehmend auch Unternehmen in die Entwicklung quelloffener Software involviert sind und in diesem Zusammenhang der Community auch Teile ihres Patentportfolios zur Verfügung stellen[19].

Wirft man einen Blick auf die mit der Beantragung eines Patents einhergehenden Kosten, ist aber auch nachvollziehbar, dass diese selbst für viele Unternehmen eine Belastung darstellen: Alleine die Verwaltungskosten für ein „durchschnittliches Patent" belaufen sich in der Europäischen Union auf etwa 28 500 Euro. Insofern stellen Softwarepatente nicht nur ein Problem für die Gemeinde der quelloffenen Softwareentwickler dar. Diese hat allerdings den strategischen Nachteil, dass die Überprüfung auf Patentverletzungen durch die Zugriffsmöglichkeit auf den Quellcode sehr viel einfacher ist als im Fall proprietärer Software.

Bei näherer Betrachtung ist das Problem der Patentierung von Software aber nicht alleine in den damit verbundenen Kosten zu sehen. Vielmehr gilt es, die sich aus den § 9-13 PatG ergebenden Verbotsrechte zu beachten. Danach kann ein Patentinhaber Dritten die Nutzung einer bestimmten Erfindung untersagen oder Schadensersatz im Fall der unberechtigten Nutzung einfordern. Dabei gilt es aber zwischen einer privaten und gewerblichen Nutzung zu differenzieren. Im ersten Fall darf nach § 11 PatG eine Erfindung genutzt werden, ohne dass Lizenzgebühren an den Patentinhaber zu entrichten sind. Von einem Hobbyentwickler kann in der Regel auch nicht erwartet werden, dass er umfangreiche Patentrecherchen durchführt, weshalb ihm eine Verletzung der Sorgfaltspflicht nur schwer nachzuweisen ist. Anders sieht das aus, wenn quelloffene Software für gewerbliche Zwecke genutzt oder durch ein Unternehmen vermarktet wird. Die Weiterentwicklung vorhandener Programme kann in diesem Fall durch Forderungen aus Patenten oder Schadensersatzansprüchen stark beeinträchtigt werden. Auch zukünftig ist mit einer Änderung dieser Situation nicht zu rechnen: In den vergangenen Jahren sind die Patentanmeldungen im Bereich der computerimplementierten Erfindungen kontinuierlich gestiegen.

### Fazit

Ob es einem gefällt oder nicht: „Softwaretechnik ist auch in Europa schon lange kein patentfreier Raum mehr – und wird es auch nicht."[20] Die Ablehnung der Softwarepatentrichtlinie als einen Erfolg zu feiern, erscheint insofern zweifelhaft, als mit dem Erhalt des Status quo der Entwicklungsprozess quelloffener Software nach wie vor durch Patente behindert wird. In diesem Zusammenhang ist die Aussage des CSU-Europaabgeordneten Joachim Wuermeling zu sehen, wonach sich die Open-Source-Bewegung „selbst ins Knie geschossen" habe[21]. Aber auch für die Unternehmen der deutschen Softwarebranche – die durch kleine und mittelständische Unternehmen geprägt ist – stellen die damit einhergehenden Kosten eine erhebliche Belastung dar.

Mit dem Scheitern der Softwarepatentrichtlinie wurde somit – entgegen der weit verbreiteten Auffassung – kein Votum gegen die Vergabe von Softwarepatenten abgegeben. Wirft man einen Blick in das Sondierungspapier der Europäischen Kommission, haben letztendlich die Befürworter der Richtlinie ihr Ziel erreicht: Eine Patentvergabe, die sich an den Vorgaben des Europäischen Patentamts orientiert[22]. Dieser Ausgang erscheint problematisch, da der Ausschlusstatbestand für Software von der Patentierbarkeit unklar und in der Praxis kaum von Bedeutung ist. Die Rechtsunsicherheit bezüglich der Patentierbarkeit computerimplementierter Erfindungen bleibt daher vorerst erhalten.

---

[18] Vgl. A. Horns: Der Patentschutz für softwarebezogene Erfindungen im Verhältnis zur „Open Source"-Software, in: Internet-Zeitschrift für Rechtsinformatik, 2000, Abs. 1-80, http://www.jurpc.de/aufsatz/20000223.htm.

[19] Vgl. C. Maaß, E. Scherm: Wettbewerbsanalyse des Softwaremarktes – Der Einfluss von Open-Source-Software, in: Zeitschrift Führung + Organisation, 74. Jg. (2005) (in Drucklegung).

[20] Vgl. C. Schulz: Brüsseler Perspektiven, in: Linux Magazin, 2003, Nr. 12, S. 64-66.

[21] Vgl. W. Herrmann, a.a.O.

[22] Vgl. Europäische Kommission, a.a.O.