

Dieser Artikel ist Teil des
Open Source Jahrbuch 2005



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

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

Von Lizenzen und Patenten

KATJA LUTHER



(CC-Lizenz siehe Seite 463)

1. Einleitung zum Kapitel Recht und Politik

Bei dem Begriff Open Source denkt man in erster Linie an Softwareentwicklung und evtl. noch an ein Vertriebskonzept für Software, aber wahrscheinlich nicht an Recht und Politik. Doch vor allem im letzten Jahr konnte man mehrfach erfahren, welchen Einfluss Entscheidungen der Politik oder Urteile von Gerichten haben können.

In diesem Kapitel wird dieser Einfluss aus der Sichtweise von Juristen (Thomas Ebinger und Andreas Neumann) sowie eines Journalisten (Stefan Krempf) betrachtet.

2. Softwarepatente – all überall

„Ein Start-up mit keinen eigenen Patenten wird künftig gezwungen sein zu zahlen, was die Größen zu verlangen belieben und der Preis könnte hoch sein – etablierte Firmen haben ein Interesse daran, künftige Wettbewerber auszugrenzen.“¹

Durch die Debatte über Softwarepatente wurde im letzten Jahr in breiterer Öffentlichkeit als zuvor auch über Softwarelizenzen und Open Source diskutiert. Man erkennt die Brisanz dieses Themas auch an dem Anteil, den es in diesem Kapitel hat. Bisher fiel Software unter das Urheberrecht, wodurch zwar der Quelltext an sich geschützt wurde, nicht aber die Idee, die dahinter steckte. Die Implementierung eines alternativen Programms zur Bearbeitung eines Problems, das bereits durch ein ähnliches Programm gelöst wurde, verletzt kein gültiges Recht. Das Urheberrecht schützt also die konkrete Ausdrucksform einer Lösung, Patente hingegen eher die Lösungsidee an sich.

Patente wurden als Anreiz zum Schaffen neuer Erfindungen eingeführt. Nach der Erteilung des Patentbesitzes muss die Erfindung in der Patentschrift veröffentlicht werden, der Patentinhaber erhält aber das absolute Recht an der Erfindung. In der Softwareentwicklung ist es jedoch zu bezweifeln, dass die Patentierung eines Programmes

¹ Bill Gates zitiert in Sietmann (2002)

oder eines Algorithmusses zu einem Innovationsanreiz führt. Zum einen ist die Entwicklung in der Softwareproduktion so schnell, dass eine patentbedingte Blockierung eines Algorithmusses oder eines Computerprogrammes eher zu einer Bremse der Innovation führt bzw. große Firmen stark bevorzugt. Nur sie sind in der Lage, die kostenintensive Patentrecherche und die anfallenden Patentreizen zu bezahlen. Zum anderen liegt es nahe, dass die freie Weiterentwicklung existierender Lösungen auch eher zu Innovationsförderung führt, anstatt dass jeder das Rad neu erfinden muss.

Bei der Vergabe von Patenten gelten feste Regeln, wann einer Erfindung ein Patent zusteht:

Anhand der Beschreibung muss ein Fachmann in der Lage sein, die Erfindung nachzubauen, die Erfindung muss neu sein und eine gewisse Erfindungshöhe aufweisen, das heißt, dass nicht jeder andere bei der Lösung desselben Problems zur gleichen Lösung kommen würde. Außerdem muss die Erfindung einen wirtschaftlichen Nutzen haben. Patente sichern dem Erfinder für eine gewisse Zeit die Rechte an seiner Erfindung.

In Europa wurde durch das Europäische Patentübereinkommen (EPÜ) 1973 das Patentrecht einheitlich geregelt. Es wurde auch definiert, welche Erfindungen nicht durch Patente geschützt werden können. In Artikel 52(2) EPÜ werden „Programme für Datenverarbeitungsanlagen“ ausdrücklich von der Patentierbarkeit ausgenommen. Schon 1985 wurde diese Formulierung aufgeweicht, indem man statt der Software als solche nur noch nicht-technische Software von der Patentierbarkeit ausschloss. Nun war es an den Patentrichtern zu entscheiden, wann ein Programm technischer Natur war und wann nicht. Diese Entscheidung konnte natürlich sehr unterschiedlich ausfallen. Durch diese rechtliche Lücke wurden in den letzten Jahren in Europa schon etwa 30 000 Softwarepatente erteilt.²

Um wieder eine einheitliche Linie in das europäische Patentrecht zu bekommen, entwarf die Kommission der Europäischen Union 2002 eine neue Richtlinie zu „computerimplementierten Erfindungen“. Diese wurde dann zunächst vom Europäischen Parlament diskutiert. Am 24. September 2003 legte das Parlament seinen Standpunkt fest, in dem der Schwerpunkt wieder stärker auf die „Technizität“ einer Erfindung gelegt wurde. Dadurch wurde eine stärkere Einschränkung der Patentierbarkeit von Software gefordert als in der Richtlinie der Kommission.

Der Ministerrat der EU wiederum machte 2004 ebenfalls einen Vorschlag für die Richtlinie über Patentierbarkeit „computerimplementierter Erfindungen“. Dieser geht wieder in die Richtung der Richtlinie der Kommission und erlaubt nahezu jegliche Patentierung von Software. Die Richtlinie des Rates führte bei vielen Interessensverbänden zu Protest. Nicht nur die Open-Source-Gemeinde fürchtet, nicht mehr ohne Patentverletzungen programmieren zu können, auch kleine und mittelständische Unternehmen fürchten um ihre Existenz. Sie argumentieren, dass nur große Softwareunternehmen von dieser Regelung profitieren werden, da sie zum einen häufig schon jetzt große Mengen an Patenten besitzen, die sie unentgeltlich nutzen können und zum anderen haben sie die finanziellen Mittel, um bei der Entwicklung für die

2 <http://swpat.ffi.org/patente/zahlen/>

nötige Patentrecherche und evtl. anfallende Patentgebühren aufzukommen. Da die EU einem Monopol oder Oligopol entgegenwirken sollte, kann die Verabschiedung der Richtlinie des EU-Rates eigentlich nicht im Interesse der EU sein.

Im Dezember 2004 sollte der Rat über diese Richtlinie abstimmen, doch durch das Votum Polens, das noch Beratungsbedarf sah, wurde die Verabschiedung noch einmal verschoben. In der Bundesrepublik Deutschland verabschiedete der Bundestag Ende Januar 2005 einen Antrag, der sich ausdrücklich für eine „effektive“ Begrenzung der Patentierbarkeit von Software aussprach und die Justizministerin Zypries darauf hinwies, dies auch im EU-Rat durchzusetzen (Krempf 2005a). Obwohl die Bundesregierung anfangs noch den Standpunkt des Parlaments teilte, unterstützte sie bis zum Votum Polens die Richtlinie des Rates.

Der letzte Stand bis zum Redaktionsschluss war, dass das EU-Parlament eine Rekonsultation der Debatte über die Patentierbarkeit „computerimplementierter Erfindungen“ bei der EU-Kommission beantragte, der weitere Verlauf jedoch noch nicht geklärt wurde (Krempf 2005b). In dem Artikel von Stefan Krempf wird die Entwicklung der Richtlinie zu Softwarepatenten genauer betrachtet und auch die Auswirkungen auf die Softwarebranche analysiert.

3. GPL – rechtskräftig

Neben der Diskussion um Softwarepatente gab es im letzten Jahr eine wichtige Entscheidung des Münchner Landesgerichts zu der Rechtsgültigkeit der Open-Source-Lizenz *General Public Licence* (GPL). Einleitend zunächst ein paar Worte zu Open-Source-Lizenzen allgemein. Der Open-Source-Ansatz steht den Befürwortern von Softwarepatenten genau entgegen. Die Open-Source-Bewegung ist davon überzeugt, dass gerade Software, deren Quelltext offen und frei verfügbar ist, Innovation fördert und dazu motiviert, neue Lösungen zu finden. Die weitere Verwertung von Open-Source-Software wird mittels zugehöriger Lizenzen geregelt. Diese Open-Source-Lizenzen legen fest, welche Rechte, aber auch welche Pflichten der Verwerter der Software besitzt. Eine Lizenz wird als Open-Source-Lizenz anerkannt, wenn sie die von der Open-Source-Initiative geforderten Richtlinien erfüllt.³

Trotz dieser Richtlinien gibt es noch sehr unterschiedliche Lizenzen. Zum Beispiel im Hinblick auf die Weiterverwertung der Software oder von Teilen des Quellcodes sehen die Lizenzen verschiedene Pflichten für den Nutzer vor.

Die beiden bekanntesten Lizenzen sind die GPL und die BSD (Berkeley Software Distribution). Außer bezüglich der Richtlinien, die von Open-Source-Lizenzen gefordert werden, unterscheiden sich diese beiden Lizenzen sehr stark. Die am weitesten verbreitete Open-Source-Lizenz ist wahrscheinlich die GPL, die 1989 von Richard Stallman und Eben Moglen ins Leben gerufen wurde. Software, die unter der GPL⁴ veröffentlicht wird, darf ein Nutzer nur weiterverbreiten, wenn dies auch wieder unter den Bedingungen der GPL geschieht. Das Prinzip, dass jede weiterentwickelte Soft-

3 <http://www.opensource.org/docs/definition.php>

4 <http://www.fsf.org/licenses/gpl.html>

ware, die auf einer GPL lizenzierten Software basiert, auch wieder unter der GPL lizenziert werden muss, nennt man „Copyleft“⁵. Der Begriff Copyleft wurde von den Entwicklern der GNU-Lizenz geprägt. Ein Programmierer nutzt also zunächst sein „Copyright“ (Urheberrecht), um seiner Software Vertriebsbestimmungen hinzuzufügen, die jedem erlauben, das Produkt weiterzuverbreiten und zu verändern. Dann legt der Programmierer mit der „Copyleft“ genannten Voraussetzung fest, dass jede Weiterverbreitung oder Veränderung auch wieder unter der GPL stehen muss.

Der Name der BSD-Lizenz stammt von einer UNIX-Distribution der Berkeley Universität⁶, die als Open Source herausgegeben wurde. Der Lizenztext ist sehr kurz⁷ und beschränkt die Weiterverbreitung nur dahin gehend, dass die Lizenz beigefügt werden muss. Der wichtigste Unterschied zur GPL ist, dass der Lizenznehmer die Software auch im Binärcode und kommerziell verbreiten kann, es existiert keine Pflicht zur Weitergabe des Quellcodes.

Bisher gab es keine gerichtliche Entscheidung darüber, ob die GPL eine gültige Softwarelizenz ist (Kanzlei Dr. Bahr 2005). Vor dem Münchner Landgericht wurde im März 2004 über die Frage entschieden, ob die GPL in Deutschland gültig ist. In dem Artikel von Thomas Ebinger wird dieser Prozess dokumentiert. Er beschreibt, wie es dazu kam, dass ein deutscher Open-Source-Entwickler gegen eine Firma einen Unterlassungsanspruch geltend machte.

Die angeklagte Firma hatte auf ihrer Webseite Software zum Download angeboten, die ein Modul einer unter der GPL veröffentlichten Software enthielt. Dies war ohne einen Verweis auf die GPL oder einen *Link* auf den Lizenztext geschehen. Das Landgericht München bestätigte nun, dass die GPL als Allgemeine Geschäftsbedingungen gilt und vom Nutzer der Software auch eingehalten werden muss. Was dies bedeutet, hat Thomas Ebinger in seinem Artikel genauer analysiert.

Ein Problem der GPL in Deutschland kann der generelle Haftungsausschluss sein, der aus dem amerikanischen Urheberrecht kommt. Nach deutschem Recht gilt er jedoch nicht. Dadurch kann es in Deutschland dazu kommen, dass keinerlei Haftungsausschluss geregelt ist. Ein Programmierer könnte dann schon bei kleinen Fehlern bzw. für einfache Fahrlässigkeiten haftbar gemacht werden (Krempel 2004, Jaeger 2004). Unter anderem aus diesem Grund wurde in Bremen eine neue Open-Source-Lizenz erstellt, die sich an dieser Stelle von den bisherigen Lizenzen unterscheidet. Sie beschränkt die Haftung und Gewährleistung ausdrücklich auf grobe Fahrlässigkeit und Vorsatz.⁸

4. Die Politik und ihr Verhältnis zu Open Source

Beschäftigt man sich mit Recht und Politik in Bezug auf Open Source, kam man im letzten Jahr an den zwei schon erwähnten Entwicklungen rund um Softwarepatente

5 <http://www.gnu.org/copyleft/copyleft.html>

6 <http://www.berkeley.edu/>

7 <http://www.opensource.org/licenses/bsd-license.php>

8 http://www.osci.de/bibliothek/bremer_lizenz.pdf

und Open-Source-Lizenzen nicht vorbei. Darüber hinaus sollte man die allgemeine Einstellung der Europäischen Union bzw. der deutschen Politik bezüglich Open Source betrachten.

4.1. Europäische Union

Für den europäischen Softwaremarkt, der im Gegensatz zum amerikanischen vor allem aus kleinen und mittleren Betrieben besteht, sind die Ziele der Europäischen Union von ähnlich großer Bedeutung wie die Ziele der einzelnen Länder. Dies resultiert aus der Tatsache, dass Software im Allgemeinen nicht an Ländergrenzen gebunden ist. Aus diesem Grund ist der Blick auf die Ziele der EU in Hinblick auf Open-Source-Software von besonderer Bedeutung. Andreas Neumann analysiert die Ziele und Aufgaben der EU und inwieweit sich diese in der Unterstützung von Open-Source-Projekten widerspiegeln.

Die EU ist ursprünglich als Wirtschaftsunion gegründet worden und auch heute sind die vorrangigen Ziele der Gemeinschaft die Errichtung eines gemeinsamen Marktes und eine Währungs- und Wirtschaftsunion. Da Open-Source-Software in den allermeisten Fällen entgeltfrei zur Verfügung gestellt wird, kommt Neumann zu dem Ergebnis, dass deren Abgabe entgegen dem Konzept der Wirtschafts- und Währungsunion steht. Quelloffene Software tritt in Form kostenfreier Konkurrenzprodukte sozusagen in Konkurrenz zum herkömmlichen Wettbewerb. Allerdings fördert sie auch die vor- und nachgelagerten Märkte (wie den Hardwaremarkt bzw. den Dienstleistungsmarkt). Außerdem arbeitet Andreas Neumann heraus, dass durch die Veröffentlichung von Sourcecode auch Netzeffekte entstehen, die zu einer nicht zu vernachlässigenden Innovationsförderung führen. Doch auch im Hinblick auf Monopole und die Interoperabilität beschäftigt sich die EU mit der Förderung quelloffener Softwareentwicklung. Das Ziel der Europäischen Gemeinschaft ist, eine möglichst hohe Interoperabilität im Bereich der Informations- und Kommunikationstechnologie zu erreichen. Da große Netze einen Wettbewerbsvorteil gegenüber kleineren Netzen haben, birgt die Bildung solcher Netze die Gefahr von Monopolbildung des Netzanbieters. Um dies zu verhindern, versucht die EU eine Sicherstellung von Interoperabilität zu erreichen – unter anderem durch Berücksichtigung von Open-Source-Software. Die Förderungspolitik der EU steht im Gegensatz zu der Initiative des Ministerrates, die Richtlinie über „computerimplementierte Erfindungen“ zu verabschieden (siehe oben). In dem sogenannten „Aktionsplan eEurope“ wird die Förderung von Open-Source-Software vor allem in den Bereichen eLearning, eSecurity und eGovernment gefordert.

4.2. Bundesrepublik Deutschland

„Wir bekennen uns zu offenen Standards und zur Vielfalt in der Software-Landschaft der Behörden, weil durch den Wettbewerb um beste

Lösungen Qualität und Innovationen gefördert werden.“⁹

Auch in Deutschland gehen bereits einige Kommunen und Verwaltungen den Weg in Richtung Open Source. Das sicherlich bekannteste Beispiel ist die Stadt München mit dem LiMux-Projekt¹⁰, das im letzten Sommer kurz wegen Unsicherheiten bezüglich der eventuell entstehenden Kosten durch Patentgebühren ausgesetzt, nach wenigen Tagen jedoch wieder aufgenommen wurde (Löding 2004). Die Bundesregierung hat ähnlich wie die EU einen Aktionsplan zur Förderung von eGovernment-Lösungen: „Bund Online 2005“. Es ist geplant, bis 2005 alle internetfähigen Dienstleistungen auch im Internet anzubieten, wobei Innenminister Otto Schily darauf hinweist, dass die Bundesregierung auf Vielfalt im Bereich der Software achten wolle (Bundesministerium des Innern 2003).

Die Projektgruppe von „Bund Online 2005“ arbeitet eng mit der KBSt (Koordination- und Beratungsstelle für Informationstechnik in der Bundesverwaltung) zusammen. Durch Bereitstellung von Informationen zu Migrationen und Open-Source-Projekten fördert die KBSt die Verwendung von Open-Source-Software in der Bundesverwaltung.¹¹ Für Verwaltungen, die vor der Entscheidung stehen, ob sie ihre Software auf Open Source umstellen wollen, wurde der Migrationsleitfaden herausgegeben, der vor allem aus wirtschaftlichen Gesichtspunkten Hilfestellung bei der Entscheidung bietet. Es wurden also gute Voraussetzungen geschaffen für eine möglichst breite Software-Vielfalt in der öffentlichen Verwaltung und somit auch für den Einsatz von Open-Source-Software. In den nächsten Jahren wird sich zeigen, ob die Chancen genutzt werden. Auch im Bereich der Softwarepatente hat die Bundesregierung große Verantwortung und sollte sich dieser in der Diskussion über die kommende Richtlinie über „computerimplementierte Erfindungen“ auch bewusst sein. Eine Unterstützung der bisherigen Richtlinie würde eine Förderung der Monopole weniger großer Softwareentwickler nach sich ziehen.

Literaturverzeichnis

- Bundesministerium des Innern (2003), ‘Schily: Bundesregierung entwickelt Migrationsleitfaden für Open-Source-Software’, <http://www.staat-modern.de/>.
[http://www.staat-modern.de/E-Government/Pressemitteilungen-,10113.555145/Schily-Bundesregierung-entwick.htm?global.printview=2](http://www.staat-modern.de/E-Government/Pressemitteilungen-/10113.555145/Schily-Bundesregierung-entwick.htm?global.printview=2) [07. Feb 2005].
- Jaeger, T. (2004), ‘Ohne Verantwortung?’, <http://www.ifross.de>.
http://www.ifross.de/ifross_html/art3.html [07. Feb 2005].
- Kanzlei Dr. Bahr (2005), ‘LG München I: GPL ist wirksam’, <http://www.heyms-drbahr.de/news/>.
http://www.heyms-drbahr.de/news/news_det_20040724111658.html [07. Feb 2005].
- Krempf, S. (2004), ‘Erste Open-Source-Lizenz made for Germany’, *heise newsticker*.
<http://www.heise.de/newsticker/meldung/45764> [07. Feb 2005].

9 Parlamentarische Staatssekretärin im Bundesinnenministerium Ute Vogt auf dem Linux-Tag 2004 (Bundesministerium des Innern 2003)

10 <http://www.muenchen.de/Rathaus/dir/limux/projekt/89257/index.html>

11 <http://www.kbst.bund.de/Software/,-74/Open-Source.htm>

Von Lizenzen und Patenten

- Krempf, S. (2005a), 'Rechtsausschuss des Bundestags stimmt gegen Softwarepatente', *heise newsticker*. <http://www.heise.de/newsticker/meldung/55568> [07. Feb 2005].
- Krempf, S. (2005b), 'Softwarepatentrichtlinie: EU-Parlament verlangt Neustart des Verfahrens', *heise newsticker*. <http://www.heise.de/newsticker/meldung/55878> [07. Feb 2005].
- Löding, T. (2004), 'München legt Linux-Projekt wegen der Softwarepatente auf Eis', *heise newsticker*. <http://www.heise.de/newsticker/meldung/49735> [07. Feb 2005].
- Sietmann, R. (2002), 'Wissen ist Geld', *c't magazin für computertechnik* **24**, S. 108. <http://www.heise.de/ct/02/24/108/> [07. Feb 2005].