

Technische Universität Berlin
Fachbereich Informatik
Institut für Wirtschaftsinformatik
Prof. Dr. iur. Bernd Lutterbeck

**Expertengespräch
«Softwarepatente und Open Source»
am 21. Juni 2001
Ausschuss für Kultur und Medien des
Deutschen Bundestages**

Fragen und Antworten

Persönlicher Hintergrund

Ich bin Volljurist, arbeite aber seit 20 Jahren mit fachlichen Schwerpunkten in der Informatik, der Ökonomie und der Politikwissenschaft.

Seit gut 15 Jahren bin ich Professor für Wirtschaftsinformatik an der Technischen Universität Berlin mit dem Schwerpunkt Informationsrecht.

Seit 1995 bin ich zusätzlich «Jean Monnet Professor» für Europäische Integration.

Vor dieser Tätigkeit war ich Beamter des Bundesinnenministers mit einer Tätigkeit beim Bundesdatenschutzbeauftragten.

Meine Forschungsschwerpunkte liegen allgemein im Gebiet «Internet Governance», spezieller habe ich in den letzten Jahren Probleme behandelt aus den Bereichen Freedom of Information und Property Rights-Theorie.

Im Dezember 2000 habe ich dem Bundeswirtschaftsminister ein Gutachten übergeben, das für den Gegenstand dieser Anhörung einschlägig ist:

«SICHERHEIT IN DER INFORMATIONSTECHNOLOGIE UND PATENTSCHUTZ FÜR SOFTWAREPRODUKTE • EIN WIDERSPRUCH?»

Das Gutachten habe ich zusammen mit Robert Gehring (Diplom-Informatiker, Consultant, TU-Berlin) und Axel H. Horns (Diplom-Physiker, Patentanwalt in München) erstellt. Es ist über den Server des Wirtschaftsministeriums abrufbar:

<http://www.sicherheit-im-internet.de/news/news.phtml?nnid=588>

Ich bin politisch unabhängig, gehöre also keiner politischen Partei an. Ich bin auch sonst keinen wie immer gearteten Interessen durch irgendwelche Vereinbarungen verpflichtet.

Ich bin 56 Jahre alt, verheiratet, ein Kind.

A Allgemeine Grundlagen und Ausgangslage

1. Wie wird der Patentschutz im allgemeinen begründet?

Es gibt im wesentlichen vier Theorien:

- *Die Naturrechtstheorie,*
- *die Belohnungstheorie,*
- *die Vertragstheorie und*
- *die Anspornungstheorie.*

Durchgesetzt hat sich die Anspornungstheorie, die etwa folgende Begründung für Patente gibt: Industrieller Fortschritt ist wünschenswert, dafür sind Erfindungen notwendig; Erfindungen wären aber unzureichend, wenn Erfinder keine größeren Gewinne erwarten dürften, als sie unter Wettbewerbsbedingungen erwarten dürften. Die Anspornungstheorie geht von den Annahmen aus, daß der industrielle Fortschritt wünschenswert ist. Um es für Erfinder und ihre Geldgeber der Mühe wert zu machen, sich anzustrengen und ihr Geld aufs Spiel zu setzen, muss die Gesellschaft eingreifen und die Gewinnerwartungen erhöhen. Die einfachste, billigste und wirkungsvollste Weise, einen solchen Ansporn zu schaffen, sei die Verleihung zeitlich befristeter Monopolstellungen in Form ausschließlicher Patentrechte an Erfindungen.

2. Welche Schutzrechte gibt es für Software und welche Wirkungen haben sie?

Software wird sowohl durch das Patentrecht wie das Urheberrecht geschützt.

Das Urheberrecht soll [neben anderen Zielen] unberechtigte Kopien verhindern. Das englische Wort «Copyright» drückt den Grundgedanken des Urheberrechts auch sprachlich gut aus.

Das Patentrecht verleiht ein temporäres Monopol. Für seine Laufzeit kann der Inhaber seine Rechte gegen Jedermann durchsetzen. Das Patentrecht schützt die Funktionalität des Programms.

Einen deutlich anderen Akzent setzt eine Definition des in den USA berühmten Urheberrechtlers Paul Goldstein:

Property rights as private incentive

«The principal object is to ensure consumers a wide variety of information products at the lowest possible price. Intellectual property law seeks to achieve this end through the grant of private property rights enabling individuals and businesses to appropriate to themselves the value of the information they produce, thus giving them incentive to produce still more.»

Die folgende Graphik gibt einen Überblick:

	INVENTIONS	ART
Voraussetzung	Nicht offensichtlicher erfinderischer Schritt	Unabhängige Schöpfung
Modalität	Starker Schutz, starke formale Bedingungen, für eine kurze Zeit	Schwacher Schutz, dafür schwache Bedingungen für eine lange Zeit
Negative ökonomische Prämissen	<ul style="list-style-type: none">• Nicht patentierte Innovationen verbleiben im Wettbewerb• Nicht patentierte Innovationen sind frei für reverse engineering, «but not to steel»• Laterkeitsrecht greift nur bei Verwechslungsgefahr	<ul style="list-style-type: none">• Nicht urheber* fähige Produktionen verbleiben im Preiswettbewerb• Spezielle Kulturpolitik, die aber nicht für den allgemeinen Markt von Produkten gilt.

© Reichman 1992

3. Was unterscheidet Software von anderen Erfindungen?

Kennzeichnend für Software ist ihr dualer Charakter:

Erstens:

Programme sind Texte. Diese Eigenschaft ist aber eher unwichtig. Wichtig ist: Sie verhalten sich.

Deshalb kauft man sie oder lässt sie erstellen. Sie erfüllen eine Aufgabe. PowerPoint erfüllt die Aufgabe, eine Folie zu erzeugen, die man in einer Anhörung des

Deutschen Bundestages zeigen kann. Word 98 hilft mir, einen Text zu schaffen, den ich ablesen kann.

All dies steht im scharfen Kontrast zu einem literarischen Werk. Etwa Goethe's Faust. Das Buch interessiert mich nicht, weil es sich verhält. Es interessiert mich wegen seines künstlerischen Ausdrucks.

Zweitens:

Text und Verhalten sind voneinander unabhängig. Die gleiche Funktionalität könnte jemand erstellen, ohne dass sie das ursprüngliche Programm gesehen hat.

Drittens:

Programme sind virtuelle Maschinen, erstellt im Medium Text.

Man kann von einem dualen Charakter von Software sprechen: einer Maschine, die zufällig durch das Medium Text funktioniert. Die gleiche Funktionalität könnte als Hardware realisiert sein.

Viertens:

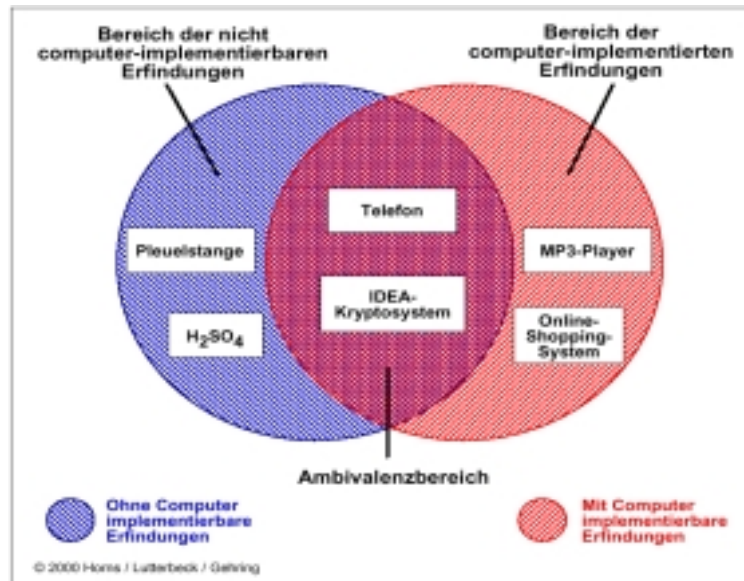
Der industrielle Design in diesen Programmen ist inkrementell in seinem Charakter, das Resultat von Methoden des Software Engineering und des praktischen Wissens der Ingenieure.

«Bringing Design to Software» ist ein bekanntes Standardwerk der Informatik. [Winograd] Klar ist, dass das Schreiben von Programmen ein industrieller Designprozess ist, nicht unähnlich dem Design von Möbeln, Küchen oder Autos.

Softwareentwicklung ist in seinem Charakter inkrementell und kumulativ. Innovation ist das Ergebnis einer Mischung von alten und neuen Elementen. Für diesen besonderen Charakter von Softwareinnovationen benutzen Informatiker gerne das Wort evolutionär. Software entsteht also in einem evolutionären Prozess unter Anwendung ingenieurswissenschaftlicher Methoden.

4. Was sind Softwarepatente, was bedeutet der Begriff der Technizität von Softwarepatenten und wie sind Softwarepatente hinsichtlich der Erfindungshöhe zu beurteilen?

Eine eindeutige Antwort lässt sich aus unserer Sicht nicht geben. In dem zitierten Gutachten an den Bundeswirtschaftsminister haben wir versucht, die Antwort auf diese Frage in einer Graphik zu verdeutlichen. Wir verwenden Begriffe, die für den «normalen» Geschmack sicher gewöhnungsbedürftig sind. Sie entsprechen aber der durch die Europäische Union vorgegebenen Sprachregelung:



Erläuterung der Grafik

In Anlehnung an die Sprachregelung des Sondierungspapiers der EU-Kommission 3 bezeichnen wir die durch den linken Kreis symbolisierte Menge als ohne Computer implementierbare Patentgegenstände. Der rechte Kreis symbolisiert diejenigen Patentgegenstände, die mit Computer implementierbar sind.

Die Schnittmenge bildet einen Ambivalenzbereich von solchen Patentgegenständen, diesowohl mit als auch ohne Computer implementierbar sind. Beispiele dafür sind etwa Erfindungen für Telefone oder Daten-verschlüsselungsgeräte, die sowohl als reine Hardwareausführung, als auch mit Softwarekomponenten konstruiert werden können.

Der linke (einfach schraffierte) Kreisausschnitt grenzt die ausschließlich ohne Computer implementierbaren Erfindungen ab. Beispielsweise fällt eine durch eine Erfindung verbesserte Pleuelstange in diesen Bereich.

Der rechte (einfach schraffierte) Kreisausschnitt umfasst symbolisch alle Patentgegenstände, die zur Implementierung eines Computers bedürfen, die computer-implementierten Erfindungen. MP3-Player, die das patentgeschützte MP3-Verfahren zur Tondatenkomprimierung benutzen, fallen beispielsweise in diesen Bereich.

Bei einem Patent lassen sich anhand der erteilten Patentansprüche allenfalls der Bereich der ausschließlich ohne Computer implementierbaren Erfindungen einerseits und der Bereich der computer-implementierten Erfindungen sicher erkennen. Bei Erfindungen im Ambivalenzbereich decken die mit dem Patent erteilten Patentansprüche sowohl reine Hardwarelösungen als auch Software-basierte Lösungen ab. Wegen dieser Unklarheiten lässt es sich nur schwer beantworten, ob ein Patent ein "Software-Patent" ist oder nicht.

Mir ist die reichhaltige Literatur um die «Technizität» von Erfindungen bekannt. Ich sehe nicht, dass diese –vornehmlich von Juristen– geführte Diskussion Entscheidendes zur Klärung der strittigen Fragen beigetragen hat. Im angelsächsischen Schrifttum hat sie entsprechend auch keine Bedeutung.

5. Unterscheiden sich Softwarepatente von Patenten für Geschäftsideen, Algorithmen und Dateiformaten und wenn ja, wie?

Das grundlegende Problem ergibt sich aus meiner Antwort zu 4.

6. Welche Unterschiede gibt es beim Schutz von Software in Deutschland im Vergleich zu anderen EU-Ländern und im internationalen Vergleich?

Es besteht ein wesentlicher Unterschied zwischen den USA und fast allen Ländern Europas. Dies ist nicht unbedingt ein Unterschied in der Rechtslage, sondern in den Strategien der Marktteilnehmer. In den USA werden Patente, aber auch das gesamte «Property Rights-Regime» aggressiv eingesetzt, um sich gegenüber dem Wettbewerber Vorteile zu sichern.

Um ein Beispiel zu geben:

Die SAP hat ihr gesamte Portfolio über Jahrzehnte ohne ein einziges Patent entwickelt und auf dem Markt durchgesetzt. Dies ist ein deutlicher Beleg, dass sich in diesem Markt die Notwendigkeit von Patenten nicht ergibt. Man braucht sie einfach nicht.

Dies ist die öffentlich geäußerte Position des Leiters der Patentabteilung von SAP, Harald Hagedorn, am 24.4.2001 in der Frankfurter Universität.

Allerdings hat SAP seine Politik 1997, also sehr spät, ändern müssen. Inzwischen erzielt SAP 30% ihres Umsatzes auf dem amerikanischen Markt –gegenüber 16% in Deutschland. Die Firma ist daher gezwungen, sich den Strategien der Mitwettbewerber auf dem amerikanischen Markt anzupassen. Sie muss daher ihre Marktposition auch durch Patente absichern.

Diese einzelwirtschaftliche Sicht ist in meiner Bewertung nicht zu widerlegen.

Aus meiner Sicht tut sich hier das wichtigste politische Problem zum Gegenstand dieser Anhörung auf: Das Problem der sog. «Softwarepatente» ist ohne Klärungen im Verhältnis USA-Europäische Union nicht möglich.

5. Nach welchen Kriterien beurteilen EPA, BGH und BPatG heute die Patentierbarkeit von Software? Sind diese Kriterien und Regeln eindeutig und welche Formulierung erweist sich bisher als am praktikabelsten?

6. Wie viele Patente für Software gibt es in Deutschland, Europa und weltweit?

Es gibt bisher keine verlässlichen Zahlen. Die verantwortlichen Beamten haben für 2000 folgende Schätzungen abgegeben:

- *Deutschland ca. 1200 (Tauchert)*
- *USA ca. 1000, einschliesslich Patente für Geschäftsmethoden.*

Es gibt wissenschaftliche Versuche, die Datenbasis des US-Patentamtes unter Zuhilfenahme der Patentklassifikation zu untersuchen. Einen solchen Versuch hat Josh Lerner, Professor an der Harvard Business School vorgenommen: [Lerner 2000, Lerner 2001] Die Auswertung der Patentklasse 705

- *Data Processing: Financial, Business Practise, Management, or Cost/Price Determination*

ergibt bei Lerner 446 Business-Patente seit 1969!

Der Direktor des amerikanischen Patentamtes nennt für 1999 die Zahl 553.

Hagedorn nennt die Zahl von 705 Business Patenten.

Es bestätigt sich meine Grundaussage: Es gibt empirisch keine verlässlichen Zahlen. Den Grund hierfür habe ich genannt: Es lässt sich nicht klar bestimmen, was ein «Softwarepatent» ist.

Noch schwieriger wird es, wenn man sich daran macht, die Zahlen «reiner» Softwarepatente zu ermitteln. Hagedorn kommt nach einer Auswertung der Patentklasse G06F auf die Zahl 12000 für den Zeitraum 1995 – 2000. Die Klasse ist folgendermassen definiert:

- **Elektronische digitale Datenverarbeitung**
Rechner, bei denen ein teil hydraulisch oder pneumatisch bewirkt wird, G06D, optisch bewirkt wird, G06E; unabhängige periphere Ein- oder Ausgabegeräte G06K; Rechnersysteme basierend auf spezifischen Rechenmodellen G06N, Impedanznetzwerke mit digitaler Technik H03H.

Zusammengefasst:

Weil die Begriffe ungenau sind, gibt es auch keine verlässlichen Zahlen.

Nach unser im Gutachten vertretenen Auffassung kann es gar kein genaueren Begriffe geben. Es kommt allerdings auch nicht darauf an.

8. Bei wie viel Prozent der bislang vom EPA gewährten 30000 Softwarepatente ist die beanspruchte Innovation so bedeutend, dass es sich für die Gesellschaft annähernd lohnen könnte, darauf ein 20jähriges Monopol zu vergeben?

Niemand sollte den Anspruch erheben, diese Frage konkret beantworten zu wollen.

B. Änderungsbedarf

9. Gibt es zwingende rechtliche Gründe (Verfassung, Internationale Verträge), die eine Neuregelung des Patentwesens auf Software erforderlich machen?

Meine Antwort fällt differenzierend aus:

Erstens:

Natürlich kann sich die Bundesrepublik zwingendem regionalen und internationalen Recht nicht widersetzen.

Man sollte aber nicht vergessen, dass Recht von politischen Akteuren nach Massgabe politischer, ökonomischer und sonstiger Interessen gemacht wird. Interessanterweise machen sich Europäer und Amerikaner wechselseitig verantwortlich, wenn sie fremde Lösungen übernehmen. So geschehen bei der Datenbank-Richtlinie, deren Inhalt der Kongress unter Berufung auf die Lösung der EU übernommen hat. So eine geläufige europäische Argumentation in der aktuellen Patentediskussion. Der führende amerikanische Informationsrechtler Lawrence Lessig von Stanford macht sich über diese europäische Haltung als «Me too patent law» lustig: Nichts spricht für die Patentierung von Software, ausser dass die Amerikaner es auch machen.

Ich verweise wieder auf meine Antwort zu 5. Die grundlegende Klärung der Rechtsprobleme muss an den europäisch-amerikanischen Wirtschaftsbeziehungen ansetzen.

Zweitens:

Die deutsche und europäische Politik müssten mögliche Spielräume in den internationalen Regularien tiefer ausloten. So erscheint es denkbar, dass die Ausnahmeregelungen der WTO-Abkommen, hier Art. 27(2) TRIPS, für eine nationale Politik in Betracht kommen könnten:

"(2) Die Mitglieder können Erfindungen von der Patentierbarkeit ausschließen, wenn die Verhinderung ihrer gewerblichen Verwertung innerhalb ihres Hoheitsgebiets zum Schutz der öffentlichen Ordnung oder der guten Sitten einschliesslich des Schutzes des Lebens oder der Gesundheit von Menschen, Tieren oder Pflanzen oder zur Vermeidung einer ernsten Gefährdung der Umwelt notwendig ist, vorausgesetzt, daß ein solchen Ausschluß nicht nur deshalb vorgenommen wird, weil die Verwertung durch ihr Recht verboten ist."

Es müsste geprüft werden, ob nicht der gesamte Bereich der IT-Sicherheit unter diese Ausnahme fallen könnte.

10. Wie beurteilen Sie die Einführung eines eigenen Schutzrechtes für Software?

Ein Schutzrecht «sui generis» hat Scharm und die Logik auf seiner Seite:

Das Dilemma der Rechtsordnung liegt nach meinen Ausführungen zu Frage 3 ja auf der Hand:

Software ist eine Maschine, die ein Verhalten erzeugt. Diese Eigenschaft schließt den Schutz des Urheberrechts aus.

Software wird andererseits in einem inkrementellen Prozess erstellt. Das führt sie weg vom Patentrecht, das ja eine bestimmte Stufe von Neuheit verlangt. Evolutionäre Softwareentwicklung und Patentrecht scheinen sich auszuschließen.

Es gibt also eine Diskrepanz zwischen den gesetzlichen Lösungen und dem hohen ökonomischen Wert von Software, der sich an den eingesetzten Investitionen messen lässt.

Eine andere Frage ist es, ob man diesem Dilemma mit einem neuen Rechtstyp begegnen will. Zumindest die deutsche Wissenschaft hat diesbezüglich keine ernsthaften Vorarbeiten gemacht. Erst nach sorgfältiger wissenschaftlicher Analyse könnte man feststellen, ob auf diesem Weg praktische Verbesserungen zu erreichen sind.

11. Wo könnte es beim Urheberrecht Unzulänglichkeiten hinsichtlich des hinreichenden Schutzes von Software vor Nachahmung geben? Wie könnte ein "maßgeschneidertes Software-Schutzrecht" aussehen?

Ich bin nicht der Meinung, das der praktisch beste Weg über das Urheberrecht geht. Dieses Rechtsgebiet passt nicht auf Software.

Der Rechtsstreit DOJ versus Microsoft zeigt aus meiner Sicht, das die grundlegenden Strukturen für diesen Markt durch Wettbewerbsrecht und wohl auch Lauterkeitsrecht eingezogen werden müssen.

12. Wie beurteilen Sie die Einführung von Sondervorschriften für Software im Patentgesetz?

Positiv.

Ein Beispiel ist das sog. Quelltextprivileg, das eine der zentralen Forderungen unseres Gutachtens zusammenfaßt:

Empfehlungen zur Patentpolitik

Der Umgang mit dem Quelltext von Computerprogrammen muss patentrechtlich privilegiert werden. Das Herstellen, Anbieten, in Verkehr bringen, Besitzen oder Einführen des Quelltextes eines Computerprogrammes in seiner jeweiligen Ausdrucksform muss vom Patentschutz ausgenommen werden. (Quelltextprivileg)

Begründung:

Diese Empfehlung muss als oberster, schlechthin unverzichtbarer Grundsatz alle Bemühungen um die künftige Patentpolitik leiten. Über den Sicherheitsgewinn hinaus setzt dieser Vorschlag die richtigen ökonomischen Anreize:

- Die Entwickler von "Open Source"-Software vermeiden das Risiko einer Patentverletzung.
- Der nichtgewerbliche Anwender beispielsweise beim privaten Gebrauch darf die von den privilegierten Entwicklern und Distributoren bereitgestellten Quelltexte ohne patentrechtliche Einschränkungen nutzen.
- Der gewerbliche Anwender fällt unter das Patent und muss für den Gebrauch des Programmes die Zustimmung des Patentinhabers einholen.
- Softwareunternehmen bekommen einen Anreiz, den Quelltext offenzulegen, um zumindest für Entwicklung und Distribution in den Genuss der Privilegierung zu kommen.

13. Macht eine Differenzierung patentierbarer und nichtpatentierbarer Güter (Erfindungen) aus Ihrer Sicht Sinn? Wenn ja, nach welchen Kriterien sollte ihrer Ansicht nach zwischen beiden Güterarten (Erfindungsarten) unterschieden werden?

Die Frage ist aus der Sicht einer politischen, mikroökonomischen und markroökonomischen Aufgabenzuweisung an das Patentwesen zu beantworten:

- *Es bestehen keine Anzeichen dafür, daß eine vollständige Entgrenzung des Patentschutzes auf alle denkbaren Gegenstände sinnvoll wäre.*
- *Das Erfordernis der Technizität für patentierbare Erfindungen sollte beibehalten werden.*

Es ist allerdings eine Illusion, Probleme in diesem Zusammenhang durch eine (Neu-)Definition oder -Interpretation des Technikbegriffes rechtssicher und praktikabel lösen zu wollen. Computerimplementierbare und -implementierte Erfindungen haben durch das Erfordernis eines Prozessors stets einen unauflösbaren Bezug zu einem physikalischen System, dessen Technizität kaum

sinnvoll in Frage gestellt werden kann.

- *Komplexe Versuche, durch gedankliche Grenzziehungen irgendwo zwischen Programm-Textform und Prozessor Probleme im Zusammenhang mit der Patentierung computerimplementierbarer oder computerimplementierter Erfindungen zu lösen, haben nicht zu befriedigenden, d.h. rechtssicheren und praktikablen Lösungen geführt.*
- *Ansätze, jegliche Erfindungen von der Patentierbarkeit auszuschließen, die auch nur rein gedanklich theoretisch computerimplementierbar sein könnten, verkennen der Rolle des Patentwesens in modernen Industriegesellschaften. Sie verkrüppeln das Patentwesen auf einen Restbereich "ölverschmierter" "Brick-and-Mortar"-Erfindungen.*

C. Auswirkungen

- 14. Welche volkswirtschaftlichen und betriebswirtschaftlichen Effekte hätte Ihrer Meinung nach eine weitestgehend unreglementierte Patentierbarkeit von sog. Softwareprodukten? Worauf basieren diese Annahmen?**
- 19. Sind Sie der Meinung, dass eine Ausdehnung des Patentschutzes für Software Innovationen in dieser Branche gefährden würden? Wie beurteilen Sie in diesem Zusammenhang die MIT Studie 'Sequential innovation, patents and imitation' von James Bessen und Eric Maskin (1999)?**

Ich bin mir über die Bedeutung der Formulierung «unreglementierte Patentierbarkeit» nicht recht im Klaren. Ich interpretiere die Frage so: Wie würden sich die Märkte unter der Annahme von Status Quo-Bedingungen weiterentwickeln?

1. Mikroökonomie

Wie aus dem SAP-Beispiel deutlich wird, müssten Unternehmen gegen ihre eigenen Interessen handeln, wenn sie auf die Einnahmen aus Patenten und ihre Marktstrategien gegen die Konkurrenten verzichten müssten. Die Patentschraube müsste sich also immer weiter drehen, da sich die Marktteilnehmer ja rational verhalten.

2. Makroökonomie

- Die Unternehmen müssten darauf dringen, dass ihre Lizenz- und Wettbewerbsstrategien durch rechtliche Rahmenregeln abgesichert werden. Der wichtigste makroökonomische Effekt wäre also ein Ansteigen der Transaktionskosten für die Durchsetzung der Patente. Diese Entwicklung zeichnet sich im Bereich des Urheberrechts bereits deutlich ab [Napster, DECSS].*
- Diese Kosten sind dem voraussichtlichen Nutzen gegenüberzustellen.*

Man sollte die Studie von Bessen/Maskin nicht überschätzen, weil sie doch einigermaßen komplizierte und deshalb fehleranfällige Modelle aufstellen. Ihre Fragestellung ist aber äußerst relevant: Lassen sich überprüfbare Indikatoren dafür aufstellen, dass durch Softwarepatente Innovationen gefördert werden. Die Autoren verneinen das. Insofern stellen sie ein mächtige These auf, die bis jetzt niemand widerlegt hat.

Man sollte diese Studie in jedem Fall mit der wissenschaftlich sehr bedeutenden Untersuchung von Kortum und Lerner aus dem Jahr 1999 zusammen lesen:

«By a process of elimination, our analysis leads us to conclude that the increase in patenting has been driven by changes in the management of innovation, involving a shift to more applied activities. Looked at from the bright side, the jump in US patenting both at home and abroad seems to indicate a real burst of innovation. The one piece of evidence to temper this rosy picture is that research investment has flattened in the 1990s.»

[Samuel Kortum, Josh Lerner: What is behind the recent surge in patenting? Research Policy 28 (1999) 1–22, <http://www.people.hbs.edu/jlerner.html>, 12.6.2001].

Diese wissenschaftliche Aussage ist auch von zentraler politischer Bedeutung: Wenn sich kein Anstieg von Innovationen –«research» ist die wichtigste Messgrösse– messen lässt, dann gibt es auch keine ökonomische Rechtfertigung für Patente.

Dies führt zu folgender Antwort auf die Frage 14:

Steigenden Transaktionskosten steht kein messbarer Nutzen gegenüber. In einer wettbewerbsgetriebenen Marktwirtschaft muss das bedeuten, dass «Softwarepatente» weder rechtlich noch ökonomisch zu rechtfertigen sind. Nach der vom Grundgesetz vorgegebenen Werteordnung muss das weiter bedeuten, dass Patente als Einschränkungen persönlicher Freiheit insoweit nicht verfassungskonform sind.

15. Welches sind die Arbeitsmethoden und wirtschaftlichen Grundlagen der Open-Source-Bewegung?

Man darf bei "Open Source" nicht nur an Software denken. Die "Open Source"-Bewegung umfasst weit mehr als ein neues Modell der Software-Entwicklung: Die "Open Source"-Bewegung entwickelt Software, Standards, Content, Geschäftsmodelle, Wissen und Ideen. Bereits die Selbstbeschreibung als "Community", wie sie in praktisch allen Aufsätzen von Akteuren selbst vorgenommen wird, weist auf die sozialen Aspekte hin.

Die am Institut für Wirtschaftsinformatik in Saarbrücken arbeitenden Ökonomen Markus Nüttgens und Enrico Tesei haben eine umfangreiche Untersuchung über Marktmodelle und das Community-Konzept von "Open Source"-Software vorgelegt. Sie fassen die Charakteristika der Communities, aus denen sich die "Open Source"-Bewegung formt so zusammen:

- Internationale Zusammensetzung der Mitglieder
- Räumlich, meist global verteilte Mitglieder

- *Internet als Kommunikationskanal*
- *Freiwilliges und unbezahltes Engagement der meisten Mitglieder*
- *Sehr wenig bezahlte Entwickler*
- *Spezielle Lizenzpolitik*
- *In bestimmten Phasen extrem hohe Entwicklungsgeschwindigkeit und -dynamik*
- *Kooperationspolitik mit kommerziellen Interessen*
- *Hohe Ansprüche an Qualität, Flexibilität und Stabilität der Software*
- *Finanzielle Unterstützung durch Spenden*

Der Kern aller dieser anderen Aktivitäten kann mit einem Stichwort beschrieben werden: Wissenstransfer. **Die "Open Source"-Bewegung orientiert sich in all ihren Facetten am Grundsatz des ungehinderten Wissenstransfers.**

Die Lizenzbedingungen für "Open Source"-Software (freie Software) formulieren diese Aussage bezogen auf Software in der Sprache der Juristen:

- «1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty [...];
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: [...]»

Die beste Charakterisierung der Tätigkeit von OSS-Entwicklern stammt in meiner Sicht von dem in Berkeley lehrenden Ökonomen Steven Weber . Bei ihm wird deutlich, dass es bei OSS um ein elementares praktisches Problem von Ingenieuren geht:

«I propose seven key principles for what people do in Open Source:

- *Pick important problems and make them interesting*
- *"Scratch an itch"*
- *Reuse whatever you can*
- *Use a parallel process to solve problems*
- *Leverage the sheer numbers*
- *Document well*
- *"Release early, release often"»*

[Steven Weber *The Political Economy of Open Source Software*, BRIE Working Paper 140. E-conomy Project Working Paper 15, June 2000, <http://brie.berkeley.edu/~briewww/pubs/wp/wp140.pdf>, 1.6.2001]

16. Welche Auswirkungen erweiterter Patentierbarkeit von Software erwarten Sie auf die Rahmenbedingungen der Softwareentwicklung unter sog. Open-Source-Lizenzen?

Der als "Freie Software" bekannte Entwicklungsprozess ist in seiner Existenz gefährdet. Betroffen sind

- *die öffentliche Kommunikation der Programmentwickler untereinander über ihre Programmierarbeit unter Nutzung des Internet sowie*
- *die Möglichkeit für jeden Interessierten, den Programmcode über das Internet zu beziehen.*

Bei einer hohen Patentdichte im Bereich der Softwareerfindungen entstehen zahllose Gelegenheiten für Patentverletzungsfälle. Die mittelständischen Akteure verfügen nicht über die Mittel, um eine aktive und/oder passive Patentpolitik betreiben zu können. [Horns 2000]

Dies ist insbesondere für den amerikanischen Markt von Bedeutung. Bei Lerner 2001 findet sich eine Zahl von etwa 3:1 – auf drei Dollar «basic research» kommt etwa ein Dollar für die rechtliche Durchsetzung von Patenten. Wegen dieser Größenverhältnisse reicht es in den USA normalerweise für den Stop von Entwicklungsaktivitäten aus, wenn einem kleinem Unternehmen ein Patentverfahren auch nur angedroht wird.

17. Würde eine Ausdehnung des Patentschutzes für Software die Open-Source-Bewegung behindern? Wenn ja, wie könnten diese Nachteile beseitigt werden?

Ja!

Als Ausweg bietet es sich an, das Patentrecht um ein Quelltextprivileg zu ergänzen – siehe die Antwort zu Frage 12.

18. Welche Auswirkungen hätte die Ausdehnung des Patentschutzes für Software auf die Entwicklung von kleinen und mittelständischen Unternehmen?

Es fehlen verlässliche Daten.

Man müsste zwischen Unternehmen unterscheiden, die Software anwenden und Software entwickeln.

Wichtig ist es jedem Fall, keine Alternative proprietäre oder quelloffene Software aufzustellen und Entwicklungen in jeder Richtung offen zu halten.

Meine Antwort auf Frage 11 gibt den Ort etwaiger Regulierungen an: das Wettbewerbsrecht.

19. Ist das Patentrecht aus Ihrer Sicht das geeignete Werkzeug, um im Softwarebereich Innovation und Fortschritt zu fördern? Sind effektivere Alternativen denkbar, und wenn ja, welche?

Eine Antwort habe ich bereits oben zu Frage 10 gegeben. Eigentlich wäre eine Lösung eigener Art erforderlich. Mangels wissenschaftlicher Vorarbeiten kann man diesen Weg aber mit gutem Gewissen nicht als Maßnahme der aktuellen Politik empfehlen.

20. Wie wirkt sich Ihrer Ansicht nach eine Klarstellung auf EU-Ebene dahingehend, dass Software und gedankliche Konzepte nicht patentierbar sind, auf bereits bestehende Patente in diesem Bereich aus?

Die Kommission denkt nicht über Klarstellungen nach, ob "Software" oder "gedankliche Konzepte" patentierbar sind. Überlegungen sind im Gange, ob auf computer-implementierte oder implementierbare Erfindungen Patente erteilt werden können.

Erteilte Patente genießen den Schutz der Eigentumsgarantie Art. 15 GG und können in ihrem materiellen Gehalt nicht ohne gesonderte verfassungskonforme Begründung angetastet werden.

Nur im Hinblick auf neue Patentanmeldungen nach einer Gesetzesänderung wären materiellrechtliche Änderungen des Schutzes möglich.

D. Eine Frage, die nicht gestellt wurde, aber gestellt werden sollte

21. Gibt es einen spezifisch deutschen oder europäischen Beitrag zur Entwicklung quelloffener Software?

Ja!

Neueste Untersuchungen von Lerner und Tirole bestätigen empirische Ergebnisse der University of North Carolina aus dem Jahr 1999 [The simple economics of Open Source, Working Paper der Harvard Business School v. 29.12.2000, <http://www.people.hbs.edu/jlerner/simple.pdf>, 1.6.2001]: Europäische Softwareentwickler sind zahlenmässig führend. Unter den Europäern stellen deutsche Entwickler mit Abstand die grösste Gruppe.

An diesen Zahlen darf die Politik nicht vorbeigehen. Es gibt offensichtlich einen spezifisch deutschen Aspekt im Thema «Softwarepatente».

Ich werde zur Anhörung neue Zahlen eines kleinen Forschungsprojektes mitbringen, die eine studentische Gruppe gerade fertiggestellt hat. Bei Bedarf könnte ich entsprechende Folien zeigen.

Berlin, den 18. Juni 2001

Prof. Dr. iur Bernd Lutterbeck