

# Patent- und Urheberrechtsschutz von Software — ein Gordischer Knoten?

Dr. Swen Kiesewetter-Köbinger\*

9. April 2003

## Zusammenfassung

Patent- als auch Urheberrecht bestimmen Inhalt und Schranken des Eigentums gemäß GG Art 14 auf ihren jeweiligen Anwendungsgebieten. UrhG §§ 69 a-g bestimmt spezifische Regelungen für den Schutz von Computerprogrammen. PatG § 1 Abs 2 3. bestimmt, dass Programme für Datenverarbeitungsanlagen nicht als Erfindungen angesehen werden. PatG § 1 Abs 3 schließt patentrechtliche Schutzbegehren auf Programme für Datenverarbeitungsanlagen als solche aus.

Eigentum, auch geistiges, hat immer etwas mit wirtschaftlichem Wert zu tun. Das Bundesverfassungsgericht hat hierzu festgestellt:

- „Das Urheberrecht ist als Nutzungsrecht „Eigentum“ iS des GG Art 14 Abs 1 S 1. GG Art 14 Abs 1 S 1 gebietet die grundsätzliche Zuordnung des wirtschaftlichen Wertes eines geschützten Werkes an den Urheber.<sup>1</sup>“
- „Es ist dabei Sache des Gesetzgebers, im Rahmen der inhaltlichen Ausprägung des Urheberrechts sachgerechte Maßstäbe festzulegen, die eine der Natur und sozialen Bedeutung des Urheberrechts entsprechende Nutzung und angemessene Verwertung sicherstellen (GG Art 14 Abs 1 S 2).<sup>2</sup>“
- „Für das Patentrecht als Eigentum im Sinne der Verfassung bedeutet die grundsätzliche Zuordnung der vermögenswerten Seite an den Patentinhaber die Freiheit, in eigener Verantwortung darüber verfügen zu können.<sup>3</sup>“

\*Der Beitrag gibt die private Meinung des Verfassers wieder.

<sup>1</sup>BVerfG 1 BvR 765/66 Leitsatz S. 1, MDR 1972, 23

<sup>2</sup>BVerfG 1 BvR 765/66 Leitsatz S. 3, MDR 1972, 23; siehe auch BVerfG 1 BvR 352/71 GRUR 1980, 44-49

<sup>3</sup>BVerfG 1 BvR 1864/95 Orientierungssatz 1b, GRUR 2001, 43-45

Eine der wichtigsten Fragen, die es hinsichtlich der Patentierung „Computer-implementierter Erfindungen“ zu klären gilt, ist demnach, wem der wirtschaftliche Wert eines Computerprogramms nach dem Willen des Gesetzgebers zugeordnet werden soll? Dem Urheber des Computerprogramms, der dieses eigenschöpferisch realisiert, oder demjenigen, der die Idee zu einem neuen Computerprogramm (Verfahren mit rein funktionaler Beschreibung) als Erster angemeldet hat, ohne dieses Computerprogramm notwendigerweise realisieren zu müssen?

## Inhaltsverzeichnis

1	Chronologie	1
2	Wessen Eigentum ist Software?	4
2.1	Äquivalenz von Hardware und Software . . . . .	5
2.2	Effektiver Patentschutz . . . . .	6
2.3	Klare und vollständige Lehre . . . . .	7
2.4	Anmeldungen ohne offenbarten Quellcode . . . . .	10
2.5	Schöpfungshöhe vs. Erfindungshöhe . . . . .	11
2.6	Technizität . . . . .	12
3	Verfassungsrechtliche Bedenken	12
4	Schlussbemerkung	14

## 1 Chronologie

In der (patentrechtlichen) Entscheidung „Dispositionsprogramm“<sup>4</sup> wurde der Begriff der Technik als das **einzig** brauchbare Abgrenzungskriterium gegenüber andersartigen geistigen Leistungen des Menschen angegeben, denen nach dem Willen des Gesetzgebers

<sup>4</sup>BGH X ZB 23/74, GRUR 1977, 96-9

andere Arten des Leistungsschutzes, insbesondere Urheberrechtsschutz, zuteil werden soll. Überschneidungen zwischen diesen verschiedenen Leistungsschutzrechten sollten nach Möglichkeit ausgeschlossen sein. Mit dieser Begründung wurde die Patentfähigkeit des beanspruchten Verfahrens, das durch ein Dispositionsprogramm ausgeführt werden sollte, bekanntlich verneint.

In der (urheberrechtlichen) Entscheidung „**Inkasso-Programm**“<sup>5</sup> wurde die Breite des Schutzbereichs begrenzt und begründet: „die wissenschaftliche Lehre und das wissenschaftliche Ergebnis sind urheberrechtlich frei und jedermann zugänglich“<sup>6</sup>. Wissenschaftlich notwendigen und technisch üblichen Ausdrucksweisen<sup>7</sup> sprach der Senat die erforderliche eigenschöpferische Prägung<sup>8</sup> ab. Seitens der Anwälte<sup>9</sup> wurde gegen diese, die Breite des Schutzbereichs einschränkende Entscheidungspraxis der Urheberrechtssenate, dass die wissenschaftliche Lehre und das wissenschaftliche Ergebnis urheberrechtlich frei und jedermann zugänglich sein muss<sup>10</sup>, massiv protestiert. Als Ausweg blieb bei dieser „inzwischen zu einem Dogma verfestigten These“<sup>11</sup> nur die Flucht weg vom Urheberrecht hin zum Patentrecht<sup>12</sup>.

In der (urheberrechtlichen) Entscheidung „**Betriebssystem**“<sup>13</sup> wurde zwar erklärt, dass Urheberrechtsschutz dem Patentschutz nicht zwingend entgegensteht<sup>14</sup>. Dem Betriebssystem, über das zu befinden war, das lediglich zur Steuerung eines Computers und seiner

Anschlussgeräte diene, wurde der technische Charakter jedoch abgesprochen<sup>15</sup>. Die reine Benutzung eines Computerprogramms wurde als urheberrechtlich nicht relevanter Vorgang bezeichnet, ebenso wie das Lesen eines Buches, das Anhören einer Schallplatte, das Betrachten eines Kunstwerks oder eines Videofilms<sup>16</sup>.

Durch das **2. Gesetz zur Änderung des Urheberrechtsgesetzes** vom 9. Juni 1993<sup>17</sup> hat der Gesetzgeber die EG-Richtlinie vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen<sup>18</sup> umgesetzt und damit einerseits gegenüber der früheren Rechtsprechung geringere Schutzanforderungen an Computerprogrammen gestellt<sup>19</sup>, andererseits aber auch nur einen engen Schutzbereich zugestanden. Eine Einschränkung auf Computerprogramme nicht technischer Natur findet sich darin nicht. Allerdings lässt UrhG § 69 g die Anwendung sonstiger Rechtsvorschriften auf Computerprogramme, insbesondere über den Schutz von Erfindungen, ... unberührt<sup>20</sup>. Anstelle der Freiheit der wissenschaftlichen Lehren und Ergebnisse wurde festgelegt, dass Ideen und Grundsätze, die einem Element eines Computerprogramms zugrunde liegen, einschließlich der den Schnittstellen zugrundeliegenden Ideen und Grundsätze, nicht geschützt sind<sup>21</sup>.

<sup>5</sup>unter Bezug auf: BPatG CuR 1988, 652, 654f; BPatG GRUR 1987, 31, 32 — „Elektronisches Übersetzungsgerät“

<sup>5</sup>BGH I ZR 52/83, GRUR 1985, 1041-1049

<sup>6</sup>unter Bezug auf: BGH I ZR 106/78, GRUR 1981, 352, 353 — „Staatsexamensarbeit“; BGH I ZR 29/79, GRUR 1981, 520, 522 — „Fragen-Sammlung“ (diese unter Bezug auf BGH I ZR 129/61, BGHZ 39, 306-312 — „Rechenschieber“)

<sup>7</sup>welche patentrechtlich zumindest naheliegend, also nicht erfinderisch wären.

<sup>8</sup>unter Bezug auf BGH GRUR 1981, 352, 353 - „Staatsexamensarbeit“

<sup>9</sup>so *Haberstumpf* „Grundsätzliches zum Urheberrechtsschutz von Computerprogrammen nach dem Urteil des Bundesgerichtshofs vom 9. Mai 1985“, GRUR 1986, 222-235 und *von Hellfeld* „Sind Algorithmen schutzfähig?“, GRUR 1989, 471-485

<sup>10</sup>BGH GRUR 1979, 464 — „Flughafenpläne“, BGH GRUR 1981, 353 — „Staatsexamensarbeit“, BGH GRUR 1981, 522 — „Fragen-Sammlung“, BGH GRUR 1984, 660 — „Ausschreibungsunterlagen“, BGH I ZR 52/83, GRUR 1985, 1041-1049 — „Inkasso-Programm“

<sup>11</sup>*Haberstumpf* Fußn. 4 a.a.O. (siehe Fußn. 9)

<sup>12</sup>also eine Flucht vor dem gesetzlichen Richter, GG Art 101 Abs 1 S 2

<sup>13</sup>BGH I ZR 139/89, GRUR 1991, 449-453

<sup>14</sup>unter Bezug auf: BGHZ 67, 22, 29 — „Dispositionsprogramm“; BGH, X ZB 19/78, GRUR 1980, 849, 850 — „Anti-Blockiersystem“; Benkard/Bruchhausen, PatG, 8. Aufl. 1988, § 1 Rdn. 104 m.w.N.

<sup>15</sup>unter Bezug auf: BPatG CuR 1988, 652, 654f; BPatG GRUR 1987, 31, 32 — „Elektronisches Übersetzungsgerät“

<sup>16</sup>Hieraus aber zu folgern, dass die Benutzung (Funktionsausübung) zusätzlich durch Patentrecht geschützt werden muss, wie dies z.B. *von Hellfeld* (a.a.O. siehe Fußn. 9) und *Tauchert* („Patentierung von Programmen für Datenverarbeitungsanlagen – neue Rechtsprechung und aktuelle Entwicklungen“ JurPC 40/2001 Abs 5, 29-31, 38) sehen, erscheint allerdings widersinnig, denn auch bei Büchern, Schallplatten, Kunstwerken und Videofilmen würde ein neben dem Besitzrecht eigenständiges Benutzungsrecht den geschützten Eigentumsgegenstand Buch, Schallplatte, Kunstwerk oder Videofilm für den, nach dem Kauf rechtmäßigen Eigentümer wertlos machen. So zumindest lässt sich das Erschöpfungsprinzip urheberrechtlich in BGH I ZR 244/97, GRUR 2001, 153-155 — „OEM-Version“ sowie BVerfG 1 BvR 775/86, GRUR 1990, 183-185 und patentrechtlich in BGH X ZB 21/94, GRUR 1998, 130-132 — „Handhabungsgerät“ sowie LG Düsseldorf 4 O 175/98, Mitt.d.P. 1999, 179-180 — „Levitationsmaschine“ verstehen.

<sup>17</sup>BGBl. I 910

<sup>18</sup>ABl. EG Nr. L 122 S. 42

<sup>19</sup>vgl. OLG München 6 U 5497/98, CR 11/1999, 688-689 — „Schutzfähigkeit von Computer-Programmen“

<sup>20</sup>TRIPS Art 13 grenzt aber Beschränkungen und Ausnahmen von ausschließlichen urheberrechtlichen Rechten auf bestimmte Sonderfälle ein, die weder die normale Verwertung des Werkes beeinträchtigen noch die berechtigten Interessen des Rechtsinhabers unangemessen verletzen.

<sup>21</sup>UrhG § 69 a Abs 2 S 2; siehe auch TRIPS Art 9 Abs 2: „Der urheberrechtliche Schutz erstreckt sich auf Ausdrucksformen und nicht auf Ideen, Verfahren, Arbeitsweisen oder mathematische Konzepte als solche.“, was die zulässige Breite des Schutzbereichs (Schranken des Eigentums) stark einschränkt.

In der (urheberrechtlichen) Entscheidung „**Buchhaltungsprogramm**“<sup>22</sup> hat der Senat zeitgleich dargelegt, nach welchen Kriterien Eigentumsrechte nach dem geänderten UrhG zugestanden werden: „Die Annahme einer Miturheberschaft setzt in rechtlicher Hinsicht ein gemeinsames Schaffen der Beteiligten voraus, bei dem jeder einen schöpferischen Beitrag leistet, der in das gemeinsame Werk einfließt<sup>23</sup>. Die schöpferische Mitwirkung kann bei einem stufenweise entstehenden Werk — wie z.B. einem Computerprogramm — auch in einem Vorstadium erfolgen, wenn sie als unselbständiger Beitrag zum einheitlichen Schöpfungsprozess der Werkvollendung geleistet wird<sup>24</sup>. Die einzelnen Miturheber brauchen nicht jeden Beitrag zum gemeinsamen Werk auch gemeinsam zu erbringen; es reicht aus, daß jeder in Unterordnung unter die gemeinsame Gesamtidee einzelne (schöpferische) Beiträge selbst erbringt<sup>25</sup>.“ Als schöpferische Beiträge wurden dabei u.a. die Gestaltung der Datenstrukturen, der Bildschirmgestaltung und auch die Struktur und die Aufteilung des Programms (RT-FIBU) mit der Aufteilung in Unterprogramme<sup>26</sup> selbst gewertet. Die Strukturierung der Daten durch verkettete Listen und Indices mit Zeigern war zwar als nicht neu und zum Standardwerkzeug beim Umgang mit größeren Datenmengen bezeichnet worden. Neu sei zum damaligen Zeitpunkt wohl nur die Verwendung dieser Datenstruktur innerhalb eines Programms zur Finanzbuchhaltung gewesen. Neuheit und das Vorliegen nicht naheliegender, eigenschöpferischer Züge wurden vom Sachverständigen und den Vorinstanzen jedoch jeweils anerkannt. Die Rechte des Schöpfers des Algorithmus werden gemäß dieser Entscheidung als schöpferische Mitwirkung zum einheitlichen Schöpfungsprozess der Werkvollendung mit berücksichtigt, wenn er zu diesem beiträgt.<sup>27</sup> Ein eigenständiges Recht z.B. auf die Zielvorgaben des

Programms (Programmidee) wird ihm dabei nicht zuerkannt.

Von ihrem technischen Aspekt her gesehen, lag der Sachverhalt bei der zwischenzeitlich ergangenen patentrechtlichen Entscheidung „**Seitenpuffer**“<sup>28</sup> mit ihren zwischengespeicherten Seitenadressen (Zeigern) der Seitenpuffer ganz ähnlich, wie bei der Strukturierung der Daten durch verkettete Listen und Indices mit Zeigern. Nach den in „Seitenpuffer“ genannten Kriterien wären die als eigenschöpferisch, also nicht naheliegend anerkannten Züge der Datenstrukturierung in dem Finanzbuchhaltungsprogramm RT-FIBU<sup>29</sup> technischer Art und damit patentrechtlich zu beurteilen. RT-FIBU ist jedoch ein typisches Anwendungsprogramm, wohingegen sich das in „Seitenpuffer“ beanspruchte Verfahren auf alle schaltungstechnische Lösungen — egal ob als fest verdrahtete Anordnung oder als programmierte Anordnung in Form von Betriebs- oder Systemsteuerprogramme verwirklicht — erstreckt. Nur bei letzterer Lösung seien Software und Hardware voneinander trennbar, argumentierte die Patentinhaberin. Betriebs- oder Systemsteuerprogrammen hatte jedoch die Entscheidung „Betriebssystem“<sup>30</sup> gerade ihren technischen Charakter abgesprochen.

Seit damals wuchs der Überschneidungsbereich, in dem für Computerprogramme sowohl urheberrechtlicher, als auch patentrechtlicher Schutz zugelassen wurde, trotz der neuen gesetzlichen Bestimmungen über den Urheberrechtsschutz von Computerprogrammen immer weiter. Unstrittig dürfte dabei wohl sein, dass urheberrechtlicher Schutz<sup>31</sup> für alle Computerprogramme gewährt wird, soweit sie das Ergebnis einer eigenschöpferischen, nicht banalen Leistung sind<sup>32</sup>. Zwischen technischen und nicht technischen Computerprogrammen wird urheberrechtlich nicht unterschieden<sup>33</sup>

<sup>22</sup>BGH I ZR 47/91, GRUR 1994, 39-41

<sup>23</sup>unter Berufung auf: BGH I ZR 174/60, GRUR 1963, 40, 41 und BGH I ZR 179/82, GRUR 1985, 529

<sup>24</sup>unter Berufung auf: BGH BGHZ 94, 276, 281 f., 284, 288

<sup>25</sup>vgl. BGH I ZR 17/58, GRUR 1959, 335, 336

<sup>26</sup>gemäß den klassischen Aufsätzen von *Parnas D.L.* „On the Criteria To Be Used in Decomposing Systems into Modules“ Comm. of the ACM, Vol. 15 No. 12 Dec. 1972, 1053-1058 und *Wirth N.* „Program Development by Stepwise Refinement“ Comm. of the ACM Vol. 14 No. 4 Apr. 1971, 221-227

<sup>27</sup>Schöpferische Tätigkeit begründet auch patentrechtlich ein (Mit-)Erfinderrecht; vgl. BGH X ZR 223/98, GRUR 2001, 226-228 — „Rollenantriebseinheit“; BGH X ZR 130/93, Mitt. 1996, 16-18 — „Gummielastische Masse“; BGH X ZR 104/90, GRUR 1994, 36-38 — „Meßventil“

<sup>28</sup>BGH X ZB 13/88, GRUR 1992, 33-36

<sup>29</sup>BGH I ZR 47/91 a.a.O. (siehe Fußn. 22)

<sup>30</sup>BGH I ZR 139/89, GRUR 1991, 449-453

<sup>31</sup>also auch Schutz des wirtschaftlichen Wertes (BVerfG 1 BvR 765/66 a.a.O. (siehe Fußn. 1), BVerfG 1 BvR 352/71 a.a.O.) der Computerprogramme.

<sup>32</sup>OLG München, 6 U 5497/98, CR 11/1999, 688-689 unter Bezug auf Schrickler/Loewenheim UrhG 2. Aufl. § 69 a Rz. 20

<sup>33</sup>OLG Hamburg, 3 U 288/00, CR 2002, 485-487 — „CT-Klassenbibliothek“; OLG Karlsruhe 6 U 52/94, GRUR 1994 726-730 — Computerprogramm für die Betriebsratsverwaltung; OLG Frankfurt 6 U 213/88 GRUR 1989 678-680; OLG München 6 U 5497/98, JurPC 30/2000 — CD-ROM Anstellersoftware; OLG München 29 U 2437/97, CR 2000, 429-431 — Datenverarbeitungsprogramm zur Verwaltung von Text- und Datenanschlüssen;

Patentrechtlich stehen diesen die Entscheidungen „Logikverifikation“<sup>34</sup>, „Sprachanalyseeinrichtung“<sup>35</sup> und „Suche fehlerhafter Zeichenketten“<sup>36</sup> gegenüber, welche als Entscheidungskriterium immer den technischen Charakter des beanspruchten Gegenstandes untersuchen. Bezüglich der parallelen Entwicklung der Entscheidungen zur Patentfähigkeit Softwarebezogener Erfindungen der Technischen Beschwerdesenate des EPA kann auf *Rees*<sup>37</sup> verwiesen werden. Die parallele Entwicklung in den USA fassen *Maier* und *Mattson*<sup>38</sup> sehr schön zusammen.

Aber auch schon vor der Novellierung des Urheberrechtsgesetzes gab es diesen Überschneidungsbereich. So wurden z.B. Computerprogramme zur „numerischen Simulation von Problemen der angewandten Mechanik, insbesondere im Zusammenhang mit dem Durchschlagen von Panzerungen, der Simulation von Explosionen und dem plötzlichen Entweichen von Flüssigkeiten“<sup>39</sup> urheberrechtlich auf Neuheit und eigenschöpferische Tätigkeit im Vergleich zu vorbekannten Programmen untersucht, um über eine urheberrechtliche Verletzungsklage entscheiden zu können. Gemäß der Entscheidung „Logikverifikation“<sup>40</sup> des X. BGH-Senats wäre dieses technische Simulationsprogramm heutzutage patentrechtlich zu beurteilen.

## 2 Wessen Eigentum ist Software?

Zur Klärung der Frage, wessen „geistiges Eigentum“ Software ist, sollte man sich zunächst einmal vergegenwärtigen, wie der Software-Entwicklungsprozess abläuft, wie die schöpferische/erfinderische Leistung der Entwicklungsphasen zu bewerten sind und wie sich patentrechtlicher und urheberrechtlicher Schutz auf die Eigentumsfrage auswirkt. Zur Verdeutlichung des Software-Entwicklungsprozesses

OLG Hamburg 3U 120/00, CR 7/2001 434-437 — Software eines FAX-Analyser-Systems; BGH I ZR 45/01, GRUR 2002 1046-1049 — „Faxkarte“

<sup>34</sup>BGH X ZB 11/98, GRUR 2000, 498-501

<sup>35</sup>BGH X ZB 15/98, GRUR 2000, 1007-1009

<sup>36</sup>BGH X ZB 16/00, GRUR 2002, 143-146

<sup>37</sup>*Dai Rees* „Some decisions of the EPO Boards of Appeal concerning software-related inventions“, Mitt.d.P. 2001, 493-501

<sup>38</sup>*Maier G.J., Mattson R.C. „State Street Bank ist kein Ausreißer: Die Geschichte der Softwarepatentierung im US-amerikanischen Recht“* GRUR Int. 2001, 677-690

<sup>39</sup>OLG Frankfurt 6 U 213/88, GRUR 1989, 678-680

<sup>40</sup>BGH X ZB 15/98, GRUR 2000, 1007-1009

ist das traditionelle Wasserfallmodell (Abbildung 1<sup>41</sup>) (als Modell der stufenweise entstehenden Werkschöpfung<sup>42</sup>) wohl am geeignets-

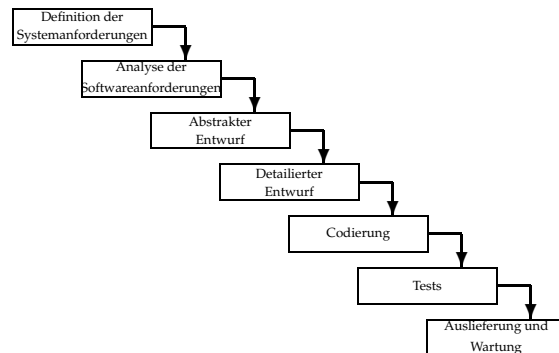


Abbildung 1: Wasserfallmodell der Softwareentwicklung

Die bei anderen gewerblichen Entwicklungen wichtige Phase der (Re-)Produktion spielt bei Software keine besondere Rolle; sie beschränkt sich auf das simple Kopieren der fertigen Computerprogramme auf digitale Datenträger.

Urheberrechtlich wird das fertige Werk, also die funktionsfähige und auslieferungsfähige Software geschützt. Da Software zumeist in Teams konstruiert wird, wird jedem Beteiligten (dem System-Ingenieur ebenso wie den Codierern) sofern er eigenschöpferische Merkmale zur Software beigetragen hat, ein Eigentumsrecht als Urheber zugesprochen<sup>43</sup>. Auch patentrechtlich entsteht das Erfinderrecht (wie das Urheberrecht) mit dem Schöpfungsakt, wenn die Erfindung **fertig**<sup>44</sup> ist. „Eine unfertige Erfindung gehört der Privatsphäre des Erfinders an und ist mit dieser geschützt.“<sup>45</sup>

In der patentrechtlichen Praxis werden regelmäßig Verfahrensansprüche oder Vorrichtungsansprüche mit vorwiegend funktionalen Merkmalen gewährt<sup>46</sup>. Schöpferische Merkmale der Codierungsphase als Beiträge zur

<sup>41</sup>aus *Brennan* „EE410 Telecommunications System Design I“ <http://www.teltec.dcu.ie/~brennanr/ee410/EE410SWEngineeringBasics.html>

<sup>42</sup>BGH I ZR 47/91, GRUR 1994, 39-41 — Buchhaltungsprogramm

<sup>43</sup>UrhG § 8 Abs 1; BGH I ZR 47/91 a.a.O. (siehe Fußn. 22)

<sup>44</sup>zu „unfertigen Erfindungen“ siehe BGH X ZB 11/90, GRUR 1992, 842-845 — Chrom-Nickel-Legierung; BGH X ZB 10/88, GRUR 1990, 510-512 — Crackkatalysator; BVerfG 1 BvR 37/63, GRUR 1964, 554

<sup>45</sup>*Schulte* PatG 6. Aufl § 6 Rdn 8; ebenso *Benkard* PatG 8. Aufl § 6 Rdn 11 und 12

<sup>46</sup>wie dies u.A. von *Tauchert* in „Patentierung von Programmen für Datenverarbeitungsanlagen – neue Rechtsprechung und aktuelle Entwicklungen“ a.a.O. (siehe Fußn. 16) propagiert wird.

Lösung des Problems werden von solchen funktional abgefassten Patentansprüchen nicht erfasst<sup>47</sup>. Patentrechtlich wird die Erfindung durch den Inhalt der Patentansprüche bestimmt<sup>48</sup>, welche alle wesentlichen (schöpferischen) Merkmale der Erfindung beinhalten müssen<sup>49</sup>. Weisen die Patentansprüche keine Merkmale der Codierungsphase auf, so steht den direkten Urhebern des Programmcodes (Codierern) kein Recht an der Erfindung zu, da die „Erfindung“ ja schon vorher fertig ist<sup>50</sup>. Zu fragen ist demnach, ob „computerimplementierte Erfindungen“ bereits durch die Angabe funktionaler Verfahrensschritte fertig sind, so dass ein Eigentumsrecht zusteht.

## 2.1 Äquivalenz von Hardware und Software

Das Argument, dass Verfahren (Algorithmen bzw. Funktionalitäten) das wesentliche Lösungsmerkmal bei der Programmentwicklung sind, weil Hardware und Software äquivalent sind, geht vermutlich auf die Argumentation von *von Hellfeld*<sup>51</sup> zurück, der die Austauschbarkeit von Hard- und Software unter Berufung auf die VICOM-Entscheidung<sup>52</sup> als Gemeinplatz titulierte und unter Berufung auf die Zukunftsvisionen des Sozialpsychologen Bammé<sup>53</sup> Algorithmen und Maschinen gleich setzt. In „Seitenpuffer“<sup>54</sup> gründete der Senat seine Entscheidung zwar nicht ausdrücklich auf diese Argumentation der Beschwerdeführerin, jedoch gewährte er ihrer diesbezüglichen Argumentation breiten Raum und sprach Anweisungen, die Elemente einer Datenverarbeitungsanlage beim Betrieb unmittelbar auf be-

<sup>47</sup>Einen Wendepunkt könnte die Entscheidung T 0049/99 (<http://legal.european-patent-office.org/dg3/pdf/t990049eu1.pdf>) der Technischen Beschwerdekammer des EPA darstellen, welche die (abstrakte) Tätigkeit eines System-Ingenieurs in der ersten Phase der Softwareentwicklung als geistige Tätigkeit beurteilt und solchen Lehren den technischen Charakter abspricht. Erst der technischen Realisierung durch den Codierer („programmer or implementation expert“) anhand der **vollständig vorliegenden** Programmbeschreibung („provided with the complete program description“), welche erst nach dem detaillierten Entwurf (vgl. Abb. 1) vorliegt, wird der technische Charakter zugesprochen.

<sup>48</sup>PatG § 14

<sup>49</sup>PatAnmV § 4 Abs 4

<sup>50</sup>vgl. *Schulte* 6. Aufl. § 6 Rdn 21 mwN

<sup>51</sup>*von Hellfeld* a.a.O. (siehe Fußn. 9)

<sup>52</sup>T 208/84, CR 1986, 193

<sup>53</sup>Bammé et. al. „Maschinen-Menschen, Mensch-Maschinen. Grundrisse einer sozialen Beziehung“, 1986, Reinbek bei Hamburg; vgl. zur Person: <http://www.uni-klu.ac.at/uniklu/org/visitenkarte.jsp?target=frame&personalnr=28>

<sup>54</sup>BGH X ZB 13/88, GRUR 1992, 33-36

stimmte Art und Weise zu benutzen, technischen Charakter und damit Patentfähigkeit zu. Ganz im Gegensatz zur Entscheidung „Betriebssystem“<sup>55</sup>, in der es hieß: „Betriebssysteme der vorliegenden Art, die lediglich der Steuerung eines Computers und der mit ihm verbundenen Anschlußgeräte dienen, stellen keine technischen Programme in diesem Sinne dar.“

In der Entscheidung „Sprachanalyseeinrichtung“<sup>56</sup> folgte der Senat anscheinend der Argumentation von der Äquivalenz von Hard- und Software, als er in seinen Gründen feststellte, dass „der Bewertungsblock und die Bevorzugungsanalyseeinrichtung sowohl durch Hardware als auch durch Software realisiert werden könne“ und erklärte in „Suche fehlerhafter Zeichenketten“<sup>57</sup>, dass in „Sprachanalyseeinrichtung“ die vorrichtungsmäßig gekennzeichneten Merkmale des zu beurteilenden Patentanspruchs, die der Lösung des Problems dienen, den technischen Charakter des damaligen Schutzbegehrens begründeten.

Auch wenn Hardware und Software gleiche Wirkungen entfalten können, so sind dies doch grundsätzlich andere Lösungen. Software in Hardware zu gießen ist schwierig und unterscheidet sich von der reinen Softwareentwicklung erheblich. Ein leicht verständlicher Einführungsartikel stammt von *Wirth*<sup>58</sup>, der erklärt, dass die rein akademische Idee, Programmcode automatisch in einen digitalen Schaltkreis zu übersetzen, außer für winzige („toy“) Programme zu einer unrealistischen Explosion der Anzahl der notwendigen Gatter führen würde. Zur Beschreibung der Unterschiede der Problemlösungsstrategien in der Hardware- und Softwareentwicklung zitiert er Chuck Thacker: „Programming [software] is about finding the best sequential algorithm to solve a problem, and implementing it efficiently. The hardware designer, on the other hand, tries to bring as much parallelism to bear on the problem as possible, in order to improve performance.“

Auch wenn gewisse Funktionsblöcke, wie in „Sprachanalyseeinrichtung“ dargelegt, theoretisch sowohl durch Hardware als auch durch Software realisiert werden können, so beruht beides jedoch auf einem fundamental anderem

<sup>55</sup>BGH I ZR 139/89, GRUR 1991, 449-453 unter Berufung auf BPatG CuR 1988, 652, 654f und BPatG GRUR 1987, 31, 32 — Elektronisches Übersetzungsgerät)

<sup>56</sup>BGH X ZB 15/98, GRUR 2000, 1007-1009

<sup>57</sup>BGH X ZB 16/00, GRUR 2002, 143-146

<sup>58</sup>*Wirth N.* „Hardware Compilation: Translating Programs into Circuits“ Computer June 1998, 25-31

intellektuellen Schöpfungsakt. Die Feststellung des BGH in „Sprachanalyseeinrichtung“ wurde vordergründig nur hinsichtlich des technischen Charakters der Anmeldung getroffen. Ob überhaupt eine fertige, nacharbeitbare (technische) Lehre<sup>59</sup> ausreichend offenbart wurde (oder nur ein unfertiges<sup>60</sup>, gewerblich noch nicht anwendbares<sup>61</sup>, außertechnisches Konzept<sup>62</sup>, welches die reine Theorie um neue Methoden bereichern soll<sup>63</sup>), wurde nicht untersucht, sondern anscheinend als gegeben angenommen, da dies bei der patentrechtlichen Prüfung wohl nicht explizit in Frage gestellt wurde.

## 2.2 Effektiver Patentschutz

In der Entscheidung „Logikverifikation“<sup>64</sup> wurde dem ausdrücklich als Programm für Datenverarbeitungsanlagen bezeichneten Computerprogramm der technische Charakter mit dem Kunstgriff zugesprochen, dass der Technikbegriff des Patentrechts nicht statisch verstanden werden kann, sondern Modifikationen zugänglich ist, sofern die technologische Entwicklung und ein daran angepasster effektiver Patentschutz dies erfordern<sup>65</sup>. Diese unterschwellige Kritik an der Ineffektivität der im Urheberrecht gewährten (und vom Gesetzgeber offensichtlich so gewollten) Rechte, wurde bereits früher u.A. von *von Hellfeld*<sup>66</sup> und *Melullis*<sup>67</sup> wesentlich deutlicher geäußert. Dreh- und Angelpunkt der Argumentation ist dabei immer der geringe Schutzbereich (also die zu engen Schranken<sup>68</sup> des Eigentumsrechtes), der idR nicht die Breite gleicher Funktionalität<sup>69</sup> abdeckt. Für *Melullis* „ist der

<sup>59</sup>vgl. BGH X ZR 43/91 „Tauchcomputer“, GRUR 1992, 430-432 unter Bezug auf BGH X ZB 18/83 „Acrylfasern“, GRUR 1985, 31-33

<sup>60</sup>vgl. BGH X ZB 10/88, GRUR 1990, 510-512 — Crackkatalysator und BGH X ZB 11/90, GRUR 1992, 842-845 — Chrom-Nickel-Legierung

<sup>61</sup>Bösl, *Teschemacher* „Die unfertige Erfindung — ein Problem der gewerblichen Anwendbarkeit?“ BIOforum 2001, 558-559

<sup>62</sup>vgl. *Melullis* „Zur Patentfähigkeit von Programmen für Datenverarbeitungsanlagen“, GRUR 1998, 843-853

<sup>63</sup>Schulte, 6. Aufl. §5 Rdn 7 unter Bezug auf RG PatBl 1889, 209 — Kongorot; BGH GRUR 72, 80 (B4b) — Trioxan

<sup>64</sup>BGH X ZB 11/98, GRUR 2000, 498-501

<sup>65</sup>unter Bezug auf BGH X ZB 15/67 „Rote Taube“, GRUR 1969, 672-674

<sup>66</sup>*von Hellfeld* a.a.O. (siehe Fußn. 9)

<sup>67</sup>*Melullis* „Zur Patentfähigkeit von Programmen für Datenverarbeitungsanlagen“ GRUR 1998, 843-853

<sup>68</sup>GG Art 14 Abs 1 S 2

<sup>69</sup>Überlegungen zur Problematik der Eigentumsverletzung von Software bei gleicher Funktionalität im US-Fallrecht bespricht *Davis* in „War of the Words: Intellectual Property Laws and Standardisation“ IEEE Micro, Dec.

Rückgriff auf das Urheberrecht unzureichend, weil die dort gewährten Rechte in der Sache hinter dem (schutzwürdigen) Bedarf der beteiligten Kreise zurückbleiben<sup>70</sup>.“ Nur die urheberrechtliche Einschränkung des Schutzbereiches auf die Programminhalte macht für *Melullis* „die für technische Sachverhalte ungewöhnliche Schutzdauer“<sup>71</sup> erklärlich.

Aber welchen wirtschaftlich<sup>72</sup> bedeutenden Unterschied macht es für einen Programmentwickler, ob er sein urheberrechtlich ihm gehörendes Computerprogramm bereits nach 20 Jahren patentrechtlich<sup>73</sup> straffrei<sup>74</sup> in Verkehr bringen darf oder erst 70 Jahre nach dem Tod eines anderen Urhebers<sup>75</sup>. Die Nutzungsdauer von Computerprogrammen<sup>76</sup> übersteigt nur in Ausnahmefällen 20 Jahre<sup>77</sup>. Nennenswerte Verkaufszahlen gibt es zumeist nur in den ersten paar Jahren. Nach zwanzig Jahren wird solch ein Computerprogramm<sup>78</sup> kaum mehr von wirtschaftlichem Wert sein und dem Wohle der Allgemeinheit dienen. Kompatible Computerhardware wird nach 20 Jahren nicht mehr erhältlich sein<sup>79</sup>. Ob die Erweiterung des Schutzbereiches im Patentrecht auf Programmideen für urheberrechtlich zu bewertende Computerprogramme, also außerhalb des einfachen Rechts<sup>80</sup>,

1993, 19-27 unter Bezug auf *Apple Computer, Inc. v. Microsoft Corp.P., Autoskill, Inc. v. National Educational Support Systems, Inc., Sega Enterprises, Ltd. v. Accolade, Inc., Computer Assoc. Int'l, Inc. v. Altai, Inc., Computermax v. UCR, Nintendo of America, Inc. v. Atari Games, Inc., Lotus Dev. Corp. v. Borland Int'l, Inc., Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc. und Plains Cotton Coop. Assoc. v. Goodpasture Computer Service, Inc.* Wieso bei patentrechtlichen Eigentumsverletzungen bei identischen Eigentumsgegenständen unterschiedlich breite Auslegungen von Inhalt und Schranken des Eigentums gelten sollen, ist nicht verständlich.

<sup>70</sup>*Melullis* a.a.O. 843

<sup>71</sup>*Melullis* a.a.O. 853

<sup>72</sup>BVerfG 1 BvR 765/66, MDR 1972, 23: „GG Art 14 Abs 1 S 1 gebietet die grundsätzliche Zuordnung des wirtschaftlichen Wertes eines geschützten Werkes an den Urheber.“

<sup>73</sup>PatG §§ 9, 10

<sup>74</sup>PatG § 142

<sup>75</sup>UrhG § 69c 3. in Verb. mit §§ 64, 65

<sup>76</sup>nicht gemeint ist hier betriebliche Individualsoftware, welche häufig zur „Altlast“ wird.

<sup>77</sup>Nach *Balzert* „Lehrbuch der Software-Technik“ Band 1, 2000, Abb. 13 erreichen diese Nutzungsdauer nur wenige Anwendungsprogramme, welche ständig an neuere Hardware und Betriebssysteme angepasst werden müssen. Welches CP/M-Programm von 1983 wird wohl heute noch nennenswert genutzt?

<sup>78</sup>z.B. für ein Fax-Analyser-System einer Faxkarte, wie es vom Hanseatischen OLG Hamburg beurteilt worden ist (3 U 120/00, CR 7/2001 434-437)

<sup>79</sup>*Hagen W.* befürchtet in „Der Stil der Sourcen“ <http://www.mikro.org/Events/OS/ref-texte/hagen/>, dass historischer Quellcode, ohne die realen Maschinen, deren Betriebssysteme und real laufende Entwicklungsumgebungen ganz unverständlich bleiben würde.

<sup>80</sup>BVerfG 1 BvR 479/92, 1 BvR 307/94, MDR 1998,216-

nicht eindeutige Entscheidungen des Gesetzgebers durch eigene rechtspolitische Vorstellungen verändert<sup>81</sup>, kann bezweifelt werden.

### 2.3 Klare und vollständige Lehre

Die Überprüfung der Vollständigkeit der offenbaren Lehre ist generell ein Problem, das gerne übergangen wird<sup>82</sup>, insbesondere bei komplexen und schwierigen Lösungen, wie sie Computerprogramme darstellen. Frühere BGH-Entscheidungen wie „Fehlerortung“<sup>83</sup> und „Tauchcomputer“<sup>84</sup> hatten jeweils explizit darauf verwiesen, dass es für die Patentfähigkeit selbstverständliche Voraussetzung ist, dass

221: „Die Rechtsfortbildung betrifft das einfache Recht, weshalb die Beantwortung der Frage, ob und in welchem Umfang gewandelte Verhältnisse neue rechtliche Antworten erfordern, ebenfalls den Fachgerichten obliegt.“;

BVerfG 1 BvR 1143/90, GRUR 1999, 226-230: „Die Auslegung des einfachen Rechts und seine Anwendung auf den einzelnen Fall sind Sache der dafür allgemein zuständigen Gerichte und der verfassungsgerichtlichen Nachprüfung grundsätzlich entzogen (vgl. BVerfGE 18, 85 <92>). Die Schwelle eines Verstoßes gegen Verfassungsrecht, den das Bundesverfassungsgericht zu korrigieren hat, ist erst erreicht, wenn die Auslegung der Zivilgerichte Fehler erkennen läßt, die auf einer grundsätzlich unrichtigen Anschauung von der Bedeutung der Eigentumsgarantie beruhen und auch in ihrer materiellen Bedeutung für den konkreten Rechtsfall von einigem Gewicht sind (vgl. BVerfGE 68, 361 <372>).“

<sup>81</sup>BVerfG 1 BvR 1186/89, MDR 1990, 895-896: „Die herkömmlichen Auslegungsmethoden und auch die analoge Anwendung einfachgesetzlicher Vorschriften sind grundsätzlich verfassungsmäßig; Schranken ergeben sich aus dem Gesetzesvorrang als Element des Rechtsstaatsprinzips bzw. der Rechtssicherheit gemäß GG Art 20 Abs 3 (hier iVm GG Art 14 Abs 1 S 1; zum Zusammenhang des eingeschränkten speziellen Grundrechts und GG Art 20 Abs 3 vgl. BVerfG, 14. Mai 1985, 1 BvR 233/81, BVerfGE 69, 315 <369ff>). Der Richter darf eindeutige Entscheidungen des Gesetzgebers nicht durch eigene rechtspolitische Vorstellungen verändern. Werden Regelungen jedoch aufgrund gesellschaftlichen Wandels oder rechtlicher Entwicklungen lückenhaft oder ergänzungsbedürftig, ist richterliche Lückensuche und Lückenschließung verfassungsrechtlich zulässig (vgl. BVerfG, 14. Februar 1973, 1 BvR 112/65, BVerfGE 34, 269 <286ff>).“

BVerfG 2 BvR 206/88, jurisweb: „Die Grenze zu der dem Gesetzgeber vorbehaltenen Gesetzeskorrektur verläuft für die zur Schließung gesetzlicher Regelungslücken notwendigen richterlichen Rechtsfortbildung dort, wo die Gerichte ohne das Vorhandensein einer sich aus Systematik und Sinn des Gesetzes ergebenden Lücke allein unter Berufung auf allgemeine, konkrete rechtliche Ableitungen nicht zulassende Rechtsprinzipien oder aus rechtspolitischen Erwägungen neue Regeln oder Rechtsinstitute schaffen.“

<sup>82</sup>Kasdan „How Courts Should Do Their Business Regarding Business Methods After *State Street Bank v. Signature Financial Group, Inc.*“ M.J. NYSBA Bright Ideas, Winter 2000 Vol.9 No. 3 <http://www.nysba.org/sections/ipl/kasdan.pdf>

<sup>83</sup>BGH X ZB 3/76, GRUR 1978,420-422

<sup>84</sup>BGH X ZR 43/91 „Tauchcomputer“, GRUR 1992, 430-432 unter Bezug auf BGH X ZB 18/83 „Acrylfasern“, GRUR 1985, 31-33

die formulierte Lehre auch in ihrem technischen Aspekt eine vollständige Problemlösung bietet. In „Fehlerortung“ wurde auf Grund der nicht vollständigen technischen Lehre der Anmeldung noch der technische Charakter abgesprochen<sup>85</sup>; in „Tauchcomputer“ wurde später klarer zwischen diesen beiden Gesichtspunkten **Technizität**<sup>86</sup> und **Vollständigkeit der Lehre**<sup>87</sup> unterschieden.

Für die Ausführbarkeit der Erfindung und die Vollständigkeit der Lehre ist es patentrechtlich bislang unerheblich, wenn es gelegentlich zu „Ausreißen“ kommt<sup>88</sup>. Gleichzeitig ist jedoch der Grundsatz zu berücksichtigen, dass bei einem Vorschlag, den ein Durchschnittsfachmann nur mit großen Schwierigkeiten praktisch verwirklichen kann, wenn er den von einem Patent angestrebten Erfolg erreichen will, keine ausreichend offenbarte technische Lehre vorliegt<sup>89 90</sup>.

Was bedeutet dies für den Software-Entwicklungsprozess?

1. Fehlerfreie Software zu entwickeln, ist eine der größten Herausforderungen mit welchen sich Software-Entwickler konfrontiert sehen<sup>91</sup>! Fehlerfreiheit lässt sich höchstens durch mathematische Korrektheitsbeweise, nicht durch Tests bewei-

<sup>85</sup>Schon bei *O'Reilly v. Morse* wurde zu breiten Ansprüchen auf „Ideen“ der technische Charakter abgesprochen. Vgl. *Kasdan* a.a.O.; *Stobbs G.A.* „Patenting Propagated Data Signals: What Hath God Wrought?“ *IEEE Communications Magazine* 7/2000 98-101; *Stern R.H.* „The PTO on software patents“ *IEEE Micro* August 1995 2-3, 77-78 und „Patenting signals“ *IEEE Micro* March/April 1998 6-8

<sup>86</sup>indirekt durch PatG § 3 Abs 1 und § 4 als notwendiges Kriterium einer Erfindung gesetzlich festgelegt.

<sup>87</sup>genau genommen regelt dies PatG § 34 Abs 4, denn nur ein ausreichend vollständig belehrter Fachmann kann die Lehre ausführen. Aber auch PatG § 5 Abs 1 soll dieses Erfordernis bezwecken (*Schulte* PatG 6. Aufl. §5 Rdn 7 unter Bezug auf RG PatBl 1889, 209 — Kongorot; BGH GRUR 72, 80 (B4b) — Trioxan).

<sup>88</sup>BGH X ZB 12/98 „Flächenschleifmaschine“, GRUR 1999, 920-923 unter Berufung auf BGH X ZR 95/87 „Sauerteig“, mwN

<sup>89</sup>BGH X ZR 3/76 „Doppelachsaggregat“, GRUR 1980, 166-169; hierzu auch BPatG 5 W (pat) 437/96 „Flächenschleifmaschine“, jurisWeb

<sup>90</sup>Welche Anforderungen an die Dokumentation der funktionalen Merkmale von Software zu stellen sind, definieren nach fachmännischem Maßstab *Parnas* und *Madey* in „Functional Documents for Computer Systems“ *Science of Computer Programming*, Vol. 25, 1995, 41-61 <http://www.crl.mcmaster.ca/SERG/papers/cr1309.pdf>; ich habe noch keine programmbezogene Patentanmeldung gesehen, welche diese Anforderungen auch nur annähernd erfüllt.

<sup>91</sup>vgl. z.B. *Giese* „Warum explodierten Mariner 1, Ariane 5, ... oder: Was kümmern mich die Probleme der Datenverarbeitung? Softwarezuverlässigkeit gestern, heute und morgen“ <http://www-aix.gsi.de/~giese/swr/>; *Brennan* a.a.O. (siehe Fußn. 41)

sen<sup>92</sup>; mathematische Korrektheit zu beweisen ist intellektuell jedoch ungleich schwieriger und anspruchsvoller. Überträgt man den in „Flächenschleifmaschine“<sup>93</sup> oder „Sauerteig“<sup>94</sup> gemeinten Verzicht auf Fehlerfreiheit auf Software, würde man „Bananen-Software“<sup>95</sup> ungerechtfertigt gegenüber sauber entwickelter Software bevorzugen.

Jedermann erwartet jedoch von einem Computerprogramm, dass es zumindest korrekt rechnen kann<sup>96</sup>. Korrektheit und Effizienz sind die Ziele, die die Arbeit des Programmierers zur Hauptsache bestimmen<sup>97</sup>.

2. Der Nachweis, dass Software den angestrebten Erfolg erzielt<sup>98</sup>, erfolgt durch die funktionale Verifikation und diese wird nach dem Wasserfallmodell (siehe Abbildung 1) in der Testphase an dem bereits entwickelten Programmcode durchgeführt. Funktionale Verifikation bedeutet den Nachweis, dass der Programmcode die Softwareanforderungen erfüllt<sup>99</sup>, also das Problem löst. Softwareanforderung umfassen dabei die Spezifikation des Verfahrensablaufs, also das was in Patentanmeldungen zu „computerimplementierten Erfindungen“ üblicherweise als Programmidee<sup>100</sup> aufgabenhaft beansprucht wird.
3. Ohne den Programmcode vorliegen zu haben, ist eine funktionale Verifikation nicht

<sup>92</sup>Dijkstra „The Humble Programmer“ CACM 15 (1972), 10: 859-866 „program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence.“

<sup>93</sup>BGH X ZB 12/98 „Flächenschleifmaschine“ GRUR 1999, 920-923

<sup>94</sup>BGH X ZR 95/87 „Sauerteig“ a.a.O.

<sup>95</sup>welche erst beim Anwender „reift“

<sup>96</sup>Das Beispiel von Giese a.a.O. zeigt sehr anschaulich, wie viel technischen Überlegungen bei der Programmierung notwendig sind, damit selbst ein einfacher Algorithmus von einem Computer mathematisch richtig ausgeführt werden kann.

<sup>97</sup>Wirth N. „Datenstrukturen und Algorithmen“ in Spektrum der Wissenschaft, 11/1984, 46-58

<sup>98</sup>BGH X ZR 3/76 „Doppelachsaggregat“ a.a.O. (siehe Fußn. 89)

<sup>99</sup>Brennan a.a.O. (siehe Fußn. 41)

<sup>100</sup>Möller „Art. 14 GG und das „geistige Eigentum““ JurPC 2002 225, Abs 55 <http://www.jurpc.de/aufsatz/20020225.htm> meint: „Einen Verfassungsauftrag, Eigentumsrechte auch auf dem Gebiet der Ideen zu schaffen oder aufrecht zu erhalten, gibt es nicht.“

Laut Davis „War of the Words: Intellectual Property Laws and Standardisation“ IEEE Micro, Dec. 1993, 19-27 kämpfte bereits Thomas Jefferson dafür, dass Ideen patentrechtlich und urheberrechtlich frei sein müssten für jedermann.

möglich; die überwältigende Mehrzahl aller softwarebezogenen Patentanmeldungen enthält jedoch keinen Programmcode<sup>101</sup>. Ohne Programmcode bedarf es zur Nacharbeit aber einer weitgehenden Neuentwicklung des Programms<sup>102</sup>, nicht nur eines einfachen Nachbaus (Kopie). Ohne Zugang zum Quellcode können idR. auch keine Fehler behoben werden<sup>103</sup>.

Die überwiegende Zahl der Patentanwälte schließt sich anscheinend der Argumentation von von Hellfeld<sup>104</sup> an, nach dem ein Informatiker jeden Algorithmus (Verfahren) programmieren kann. Glass<sup>105</sup> titulierte dies als die Sichtweise der „automatable-and-easy crowd“ und stellt diesen die Sichtweise der „complexity crowd“ gegenüber, welche die Praktiker einschließt. Als einer der wenigen, der die Frage, „Is developing software easy and automatable or complex and demanding creativity“ empirisch untersuchte<sup>106</sup>, kommt er zu dem Schluss, „that these findings carry enough significance to, once and for all, crush the automatable-and-easy side of the picture.“ Statistisch belegte die völlige Praxisferne dieser Simplifizierung die Standish Group<sup>107</sup>. Historisch zeigt Shapiro<sup>108</sup>, dass bislang die Suche nach „silver bullets“ gegen die Softwarekrise, trotz zahlloser Vorschläge zwar Verbesserungen gebracht hat, aber nie die überzogenen Versprechungen erfüllte. Dass Software schwer zu entwickeln ist, ist Lehrbuchwissen<sup>109</sup> und sollte jedem

<sup>101</sup>Nach Tauchert in JurPC 28/2002 „Zum Begriff der „technischen Erfindung““ <http://www.jurpc.de/aufsatz/20020028.htm> erfüllt Programmcode nicht das Erfordernis einer Kenntnisnahme in einer „offiziellen“ Sprache. Dem widerspricht z.B. Klopmeier „Software und softwarebezogene Erfindungen — Der Schutz von Innovationen im Spannungsfeld von Urheberrecht und Patentrecht“ [http://www.sicherheit-im-internet.de/download/SeminarSoft\\_pat.pdf](http://www.sicherheit-im-internet.de/download/SeminarSoft_pat.pdf) Seite 50

<sup>102</sup>BGH X ZR 72/98, GRUR 2002, 149-153 — „Wetterführungspläne II“

<sup>103</sup>Schneider „Neues zur Vorlage und Herausgabe des Quellcodes? Kritische Überlegungen zur Dissonanz zwischen vertraglicher und prozessualer Beurteilung des Quellcodes durch den BGH“ CR 1/2003 1-9

<sup>104</sup>von Hellfeld a.a.O. (siehe Fußn. 9)

<sup>105</sup>Glass R.L. „A Story about the Creativity Involved in Software Work“ IEEE Software 9/10 2001, 96,95 unter Bezug auf „Software Creativity“, PrenticeHall 1995

<sup>106</sup>Glass R.L., Vessey I. „Software Tasks: Intellectual, Clerical ... or Creative?“ Proc. 27th Ann. Hawaii Int. Conf. on System Science, 1994 377-382

<sup>107</sup>The Standish Group „Chaos Report“ <http://standishgroup.com/visitor/chaos.htm>

<sup>108</sup>Shapiro „Splitting the Difference: The Historical Necessity of Synthesis in Software Engineering“ IEEE Annals of the History of Computing, vol. 19, No. 1, 1997, 20-54

<sup>109</sup>Balzert „Warum ist Software so schwer zu entwickeln?“ in „Lehrbuch der Software-Technik“ 2000

Durchschnittsfachmann bekannt sein. Dass die Implementierung selbst einer einfachen Personenzugangskontrolle bei der Verwendung modernster Software-Technik und Vorgehensmodellen<sup>110</sup> erheblichen Entwicklungsaufwand verursacht, ohne die Zuverlässigkeitsanforderungen tatsächlich erfüllen zu können, zeigen *Smidts et al.*<sup>111</sup> beispielhaft.

Welche Schwierigkeiten bei der Softwareerstellung tatsächlich zu überwinden sind, ignoriert *von Hellfeld*. *Turing's* theoretischer Beweis, auf den sich *von Hellfeld* beruft, dass jeder Algorithmus im Prinzip sowohl in Hardware, als auch in Software realisiert werden kann, war ein Negativbeweis, dass ein allgemeiner Algorithmus zur Lösung mathematischer Fragen nicht gegeben werden kann<sup>112</sup>. Über die realen Schwierigkeiten der Realisierung<sup>113</sup> mittels real existierender Maschinen sagt dieser theoretische Beweis gar nichts aus. „All zu oft bleibt ein einfacher Rechner, der etwas Besonderes zu sein vorgibt, im „*Turing-Sumpf*“ stecken<sup>114</sup>. Abstrakte Turing-Maschinen sind nicht real; sie können als Gedankenexperiment nur gedacht, nicht gewerblich hergestellt oder benutzt werden<sup>115</sup>. Computerprogramme müssen hingegen die konkreten Eigenschaften physikalischer Maschinen berücksichtigen<sup>116</sup>.

Von Patentanwälten wird unter Berufung auf *von Hellfeld*<sup>117</sup> häufig behauptet, dass der Programmcode nicht offenbart werden muss, da dies der Fachmann programmieren kann<sup>118</sup>.

<sup>110</sup>Durchschnittliches fachmännischem Können ist nach *Broy und Rombach* („Software Engineering“ Informatik Spektrum Dez. 2002 438-451) wesentlich niedriger anzusetzen!

<sup>111</sup>*Smidts, Huang, Widmaier* „Producing reliable software: an experiment“ *The Journal of System and Software*, 2002, 213-224 (siehe auch *Widmaier, Smidts, Huang* „Producing More Reliable Software: Mature Software Engineering Process vs. State-of-the-Art Technology?“ *Proc. Of the 2000 Int. Conf. On Software Engineering*, 88-93); dieses Ergebnis analysiert auch *Glass* in „Error-Free Software Remains Extremely Elusive“ *IEEE Software* Jan/Feb 2003, 104,103

<sup>112</sup>*Hagen W.* „Der Stil der Sourcen“ <http://www.mikro.org/Events/OS/ref-texte/hagen/> unter Bezug auf *Turing* „On Computable Numbers, With An Application To The Entscheidungsproblem“ *Proc. London Math. Soc* (2) 42 (1936 - 7), 230 - 265

<sup>113</sup>BGH „Doppelachsaggregat“ a.a.O. (siehe Fußn. 89)

<sup>114</sup>*Kay* in „Software“, *Spektrum der Wissenschaft*, 11/1984, 34-43

<sup>115</sup>PatG § 5 Abs 1; vgl. *Schulte* PatG 6. Aufl. §5 Rdn 7 mwN.

<sup>116</sup>*Shapiro* a.a.O. über *Fetzer* „Program Verification: The Very Idea“ *Comm. ACM*, Sept. 1988, vol. 31, 1057

<sup>117</sup>*von Hellfeld* a.a.O. (siehe Fußn. 9)

<sup>118</sup>z.B. *Schiurma* in „Protokoll der Expertenanhörung Expertenanhörung des Unterausschusses Neue Medien des Deutschen Bundestags zum Thema Softwarepatente/OpenSource vom 21. Juni 2001“ <http://swpat.ffii.org/penmi/bundestag-2001/>

Hierzu beruft er sich zwar auf *Wirth*<sup>119</sup> zur Bekräftigung des Maschinencharakters von Programmen; hinsichtlich der „Machbarkeit“ von Programmen widerspricht er ihm unter Berufung auf die Patentanmeldung „Programmierungseinrichtung“<sup>120</sup> aber völlig. Für *Wirth* bleibt das Programmieren eine mathematische Disziplin bei der mathematische Methoden benötigt werden, um Computerprobleme zu lösen (was wohl niemand als **naheliegende** Tätigkeit betrachten kann).

Nach *von Hellfeld* ist das Programmieren „für den Informatiker aber eher ein „**didaktisches**“ Problem, welches dadurch bedingt ist, daß der Computer (zur Zeit) Algorithmen nur in einer hochgradig in alle Einzelschritte aufgelösten Form verstehen kann, während der Mensch wesentlich „geballtere“ Zusammenhänge erfassen kann.“

Für *Wirth* ist die zweckmäßige Wahl der Datenstrukturen häufig der Schlüssel zum erfolgreichen Programmieren und sie mag einen größeren Einfluß auf die Güte eines Programms ausüben als die Einzelheiten des verwendeten Algorithmus. Obwohl *von Hellfeld* sich auf diesen Artikel von *Wirth* bezieht, liegt für ihn „der eigentlich schöpferische Akt **grundsätzlich** im Auffinden des Algorithmus“<sup>121</sup>.

Für *Wirth* kann im Bemühen gegen das Phänomen der wuchernden Komplexität anzukämpfen kein noch so modernes Werkzeug des Software-Engineering das Talent eines Programmierers zu präzisem, konstruktiven Denken ersetzen, wohingegen *von Hellfeld* behauptet, dass „die Umsetzung eines gefundenen Algorithmus in ein lauffähiges Computerprogramm **grundsätzlich** mechanisierbar“ ist. „Wäre das Programmieren ein strikt deterministischer Prozeß, der nach festen Regeln abläuft, so wäre es bereits seit langem automatisiert worden“<sup>122</sup>.“ meint dagegen *Wirth*.

Gilt *Wirth's* Aussage auch heute noch? Liest man die zahlreichen neueren Artikel der Kolumnen „SOFTWARE CONSTRUCTION“<sup>123</sup>, „LOYAL OPPOSITION“<sup>124</sup> in *IEEE Software* oder

<sup>119</sup>*Wirth N.* „Datenstrukturen und Algorithmen“ in *Spektrum der Wissenschaft*, 11/1984, 46-58

<sup>120</sup>BPatG 17 W (pat) 124/84, GRUR 1987, 354-357; DE 31 21 445 A1

<sup>121</sup>*von Hellfeld* Fußn. 38 a.a.O. (siehe Fußn. 9)

<sup>122</sup>*Wirth N.* „Systematisches Programmieren - Eine Einführung“, 5. Aufl. B.G. Teubner, Stuttgart, 1985, 120ff

<sup>123</sup>*Hunt, A., Thomas, D.* „Zero-tolerance construction“ *IEEE Software* 9/10 2002 100-102; „Ubiquitous automation“ *IEEE Software* 1/2 2002 11-13

<sup>124</sup>*Glass, R.L.* „Reuse: what's wrong with this picture?“ *IEEE Software* 3/4 1998 57-59; „The Loyal Opposition - On Design“ *IEEE Software* 3/4 1999 104-103; „Frequently forgotten fundamental facts about software engineering“ *IE-*

„THE PROFESSION“<sup>125</sup> in IEEE Computer, so ist dies eindeutig zu bejahen. Z.B. *Hunt* und *Thomas* meinen: „The art of programming lies in that nether region between the hopeful wishes of an elegant architecture and the hard grindstone of technical details“<sup>126</sup>.“ Die wesentlichen und prägenden technischen Lösungsmerkmale stecken letztlich in dem Ergebnis der Programmierung<sup>127</sup>, dem fertigen Quellcode. Das Problem ist, die Kreativität<sup>128</sup>, die hinter der Problemlösung steckt, versteht man nicht wirklich, wenn man nicht versucht, es zu implementieren<sup>129</sup>.

Wunder bewirkende Patentrezepte für die Programmentwicklung gibt es nicht<sup>130</sup> und der Glaube an solche Patentrezepte verschärft die Probleme noch<sup>131</sup>. Bei dem derzeit hohen Entwicklungstempo der „IT“ macht die „Hastigkeit der Software“ den Computer laut *Zemanek* für den Software-Ingenieur zum Quälgeist<sup>132</sup>.

## 2.4 Anmeldungen ohne offenbarten Quellcode

Die postulierte „Automatisierbarkeit“ der Codierung lässt sich im Rahmen der Patentprüfung einer Anmeldung ohne offenbarte Quellprogramme sehr einfach überprüfen. Dazu muss man nur zwei Fälle unterscheiden:

### Es gibt tatsächlich Lösungen, die der zuständige Fachmann ohne besondere Schwierig-

EE Software 5/2001 112 -111; „A story about the creativity involved in software work“ IEEE Software 9/10 2001 96-97; „Error-Free Software Remains Extremely Elusive“ IEEE Software 1/2 2003, 104,103

<sup>125</sup> *Holmes, N.* „Why Johnny can't program [computing professionals]“ Computer 12/2000 160, 158-159 „Some comments on the coding of programs“ Computer 11/2000 128, 126-127 „The great term robbery [computer jargon]“ Computer 5/2001 94-96

<sup>126</sup> *Hunt and Thomas* „Zero-Tolerance Construction“ IEEE Software 9/10 2002, 100-102; siehe auch *McConnell* „Who Needs Software Engineering?“ IEEE Software 1/2 2001 5-8, „Who cares about software construction?“ IEEE Software 1 1996, 127-128; *Brooks* „No Silver Bullet: Essence and Accidents of Software Engineering“ a.a.O.

<sup>127</sup> vgl. BoA T 0049/99 vom 5.3.2002 <http://legal.european-patent-office.org/dg3/pdf/t990049eu1.pdf> „Information model based on a physical system“, Entscheidungsgründe 8. letzter Absatz

<sup>128</sup> *Glass* „A Story about the Creativity Involved in Software Work“ IEEE Software, 9/10 2001, 96, 95

<sup>129</sup> *Knuth* in „Der Perfektionist“ c't 5/2002 190-193

<sup>130</sup> *Brooks* „No Silver Bullet: Essence and Accidents of Software Engineering“ Computer 4/1987 <http://www.virtualschool.edu/mon/SoftwareEngineering/BrooksNoSilverBullet.html>

<sup>131</sup> *Bassett* in „Software [R]evolution: A Roundtable“ Computer May 1999 48-57

<sup>132</sup> *Zemanek* „Unsere Informationstechnik: ihre Imperfektion und ihre Instabilität“ it+ti 5/2000 5-13

keiten, nach den in „Doppelachsaggregat“ (siehe Fußn. 89) aufgestellten Kriterien findet. In diesem Fall wäre es kein Problem solchen Programmcode auf Verlangen des Prüfers nachzureichen<sup>133</sup>, da derartiges Grundwissen des Fachmanns nicht zu einer unzulässigen Erweiterung des Offenbarungsgehalts der Anmeldung führt. Dieser trivialen, naheliegenden Codierungsform der Softwareanforderungen käme weder patentrechtlich erfinderischer, noch urheberrechtlich eigenschöpferischer Gehalt zu<sup>134</sup>.

### Die Lösung in Form von Programmcode, der das gewünschte Verfahren realisiert, ist nicht ohne weitere Schwierigkeiten zu codieren.

In diesem Falle würde die Nachreichung des Programmcodes den Offenbarungsgehalt der Anmeldung unzulässig erweitern (die Erfindung wäre unfertig), da der Fachmann noch schöpferisch/erfinderisch tätig werden muss. Patentrechtlich lassen sich aus dieser Erweiterung aber keine Eigentumsrechte ableiten<sup>135</sup>. Ohne diesen Programmcode ist das gewünschte Verfahren aber nicht mehr als eine noch zu lösende (Programmier-)Aufgabe<sup>136</sup>, wie sie in den obersten Stufen des Wasserfallmodells (siehe Abbildung 1) definiert wird. Nicht ausreichend offenbarte Lösungen, „die einem Fachmann auch nach dem Lesen der Beschreibung noch nicht zugänglich wären“<sup>137</sup>, sind aber kein „Eigentum“ im patentrechtlichen Sinn und dürfen nicht in den Schutzbereich eines Patentbegehrens fallen<sup>138</sup>. „Werden zu breit gefasste Ansprüche gewährt, so können Dritte nicht mehr ungehindert forschen. Für die Öffentlichkeit kann dies bedeuten, dass ihr

<sup>133</sup> Diesen Nachweis kann nach BGH X ZB 10/88 — Crackkatalysator (a.a.O.) der Patentprüfer vom Anmelder jederzeit verlangen, um der Gefahr der Anmeldung unfertiger Erfindungen vorzubeugen.

<sup>134</sup> Zu bedenken ist jedoch, dass jeder Algorithmus, selbst trivialer, ohne die Einbindung in seinen Kontext nutzlos ist (siehe *Bird* „Software Development Viewpoints“ IEEE Computer 10/2002, 8 unter Bezug auf *Eischen* „Software Development: An Outsider's View“ IEEE Computer 5/2002, 36-44 und *Berenbach* „Distinguishing Between Programming and Software Engineering“ 8/2002, 7-8)

<sup>135</sup> PatG § 38 S 2

<sup>136</sup> BGH „Acrylfasern“ a.a.O.

<sup>137</sup> *Nuss* „Aspekte der Patentierbarkeit breiter Patentansprüche nach dem EPÜ“ Amtsblatt EPA Sonderausgabe Nr. 2 2001, 66-81 [http://www.european-patent-office.org/epo/pubs/oj001/06\\_01/06\\_spe1.pdf](http://www.european-patent-office.org/epo/pubs/oj001/06_01/06_spe1.pdf)

<sup>138</sup> *Schulte* 6. Aufl. § 34 Rdn 326-337 mwN; *Benkard*, 8. Auflage, § 14 Rdn 8, unter Berufung auf RGZ 146, 273,274; 153, 47, 50; BGH GRUR 69, 534, 535 – Skistiefelverschluss; 83, 497, 498 – Absetzvorrichtung; § 38 Rdn. 9 unter Berufung auf BGHZ 72, 119, 128f

sinnvolle Weiterentwicklungen vorenthalten werden, die der Patentinhaber nicht realisiert hat.<sup>139</sup>“

Zur Unterscheidung beider Fälle ist der „Neuheitstest“ geeignet<sup>140</sup>.

Die Beurteilung der tatsächlichen Schwierigkeit der Codierung wird sich im Einzelfall als schwierig erweisen<sup>141</sup>; rein statistisch betrachtet ist bei der Softwareentwicklung jedoch generell davon auszugehen, dass die meisten intellektuellen Schwierigkeiten gerade in den letzten Entwicklungsphasen auftreten<sup>142</sup>. Selbst mit offenbarem Quellcode ist eine Nacharbeit für ein anderes Betriebssystem, eine andere Entwicklungsumgebung oder gar eine andere Programmiersprache keine einfache Sache — schon gar keine automatisierbare<sup>143</sup>! Zusätzliche oder geänderte Funktionalität in existierenden Code zu integrieren, kann selbst überdurchschnittliche Software-Ingenieure schnell vor unlösbare Probleme stellen<sup>144</sup>. Bei der Bemessung des Schutzbereiches<sup>145</sup> sollte dies im Patenterteilungsverfahren zumindest ansatzweise berücksichtigt werden.

## 2.5 Schöpfungshöhe vs. Erfindungshöhe

Gelegentlich wird argumentiert, dass urheberrechtliche Programmschöpfung von der Qualität her niedriger zu bewerten ist, als eine erfinderische Tätigkeit<sup>146</sup>, da es ein „Mehr“ an Leistung voraussetzt. Was aber auf einer eigenschöpferischen Leistung beruht, kann für den Fachmann nicht naheliegend sein. Nicht naheliegende Lösungsmerkmale begründen aber eine erfinderische Tätigkeit<sup>147</sup>. Besondere, **höhere** Anforderungen hat der Gesetzgeber im

Patentrecht nicht gestellt. Sowohl patentrechtlich, als auch urheberrechtlich ist das Können und Wissen des Durchschnittsfachmanns<sup>148</sup>, nicht des Wissenschaftlers<sup>149</sup> maßgeblich. Damit ist jedes eigenschöpferische Merkmal, dem irgendwie technischer Charakter zuzuschreiben ist, gleichzeitig geeignet, eine erfinderische Tätigkeit zu begründen.

Mit dem Blick eines Außenstehenden mag die Arbeit eines Programmierers sich als einfache Programmieraufgabe darstellen<sup>150</sup>. Fachmännisch beurteilt, ist Software-Entwicklung aber eine der kompliziertesten, komplexesten<sup>151</sup> und schwierigsten<sup>152</sup> technischen<sup>153</sup> Entwicklungstätigkeiten<sup>154</sup>, vom Pflichtenheft bis zur auslieferungsfähigen Software<sup>155</sup>.

Ein Vergleich der jüngsten urheberrechtlichen<sup>156</sup> und patentrechtlichen<sup>157</sup> Gerichtsentscheidungen bzgl. der Schutzfähigkeit von Computerprogrammen hinterlässt allerdings den Eindruck, dass sich die urheberrechtliche „Schöpfungshöhe“ nach der sehr hohen Maß-

<sup>148</sup>Leider gibt es für Software-Technik („Software Engineering“) keine verpflichtenden Qualifikationskriterien (Curricula) wie bei anderen Ingenieursberufen (vgl. *Balzert* „Lehrbuch der Software-Technik“ Band 1, 2000, Seite 20; *Parnas* „Software Engineering: An Unconsummated Marriage“ <http://www.cs.utexas.edu/users/software/1998/parnas-19981208.pdf>).

Durch die steigende Nachfragerücklage an Software-Personal (*Balzert* a.a.O. Seite 31-32) herrscht mehr das Prinzip vor „the people who *can* are encouraged to do.“ (*Bach* „Playing the Expert Game“ *Computer* Aug. 1999 99-101), so dass man in der industriellen Praxis der Software-Technik sowohl Software-Steinzeit vorfindet, als auch Software-Abteilungen, die neueste wissenschaftliche Ergebnisse einsetzen (*Balzert* a.a.O., Seite 19). Nach *Broy* und *Rombach* („Software Engineering“ *Informatik Spektrum* Dez. 2002 438-451) sind nur 5% aller deutschen Firmen auf dem minimalen Capability-Maturity-Modell-Level 2 (Ebene der Wiederholbarkeit; vgl. <http://www.projektmanagement-glossar.de/glossar/g1-0524.html>) und der Qualifikationsgrad der Mitarbeiter (Durchschnittsfachmann) ist in der Praxis unzureichend.

<sup>149</sup>*Schulte* 6. Aufl. §4 Rdn. 41 mwN.

<sup>150</sup>vgl. *Bird* „Software Development Viewpoint“ *Computer* 10/2002 8-9 und *Berenbach* „Distinguishing between Programming and Software Engineering“ *Computer* 8/2002 7-8 in Reaktion auf *Eischen* a.a.O.

<sup>151</sup>*Balzert* „Warum ist Software so schwer zu entwickeln?“ in „Lehrbuch der Software-Technik“ Band 1, 2000, 25-35

<sup>152</sup>*Glass* „A Story about the Creativity Involved in Software Work“ a.a.O.

<sup>153</sup>*Hunt, Thomas* „Ubiquitous automation“ a.a.O.

<sup>154</sup>*Eischen* a.a.O.

<sup>155</sup>vgl. Abb. 1

<sup>156</sup>OLG Hamburg, 3 U 288/00 a.a.O.; OLG Karlsruhe 6 U 52/94 a.a.O.; OLG Frankfurt 6 U 213/88 a.a.O.; OLG München 6 U 5497/98 a.a.O.; OLG München 29 U 2437/97 a.a.O.; OLG Hamburg 3U 120/00 a.a.O.; BGH I ZR 45/01 a.a.O.

<sup>157</sup>BGH X ZB 11/98 a.a.O.; BGH X ZB 15/98 a.a.O.; BGH X ZB 16/00 a.a.O.

<sup>139</sup>*Pumfrey* „Patenschutz für breite Ansprüche“ *Amtsblatt EPA Sonderausgabe* Nr. 2 2001 82-93 [http://www.european-patent-office.org/epo/pubs/oj001/06\\_01/06\\_spe1.pdf](http://www.european-patent-office.org/epo/pubs/oj001/06_01/06_spe1.pdf)

<sup>140</sup>vgl. *Schulte* 6. Aufl. § 38 Rdn 23

<sup>141</sup>vgl. z.B. *Glass* a.a.O. in Verb. mit *McConnell* „Who Needs Software Engineering?“ a.a.O.

<sup>142</sup>vgl. z.B. *Brennan* a.a.O. (siehe Fußn. 41), *Hunt and Thomas* a.a.O.

<sup>143</sup>Für *Terekhov* und *Verhoef* „The Realities of Language Conversions“ *IEEE Software* Nov./Dec. 2000 111-124 ist Übersetzung von Programmcode so einfach wie die Umwandlung von einer Wurst in ein Schwein.

<sup>144</sup>Ein solches Beispiel beschreibt *Spinellis* in „Fear of Coding, and How to Reduce It“ *IEEE Computer*, Aug. 2001 100, 98-99

<sup>145</sup>PatG § 14

<sup>146</sup>*Brinkhof* „Zur Rechtsstellung eines angestellten Programmierers“ *GRUR* 2001, 888-885

<sup>147</sup>PatG § 4

latte in „Inkasso-Programm“<sup>158</sup> und „Betriebs-system“<sup>159</sup> nur langsam auf die erheblich niedrigere patentrechtliche „Erfindungshöhe“ zu-bewegt. Ein Grund hierfür mag vielleicht dar-in liegen, dass Urheberrechtsentscheidungen prinzipiell immer im zweiseitigen Verfahren vor Gericht bestritten werden, wohingegen die überwiegende Mehrzahl an patentrechtlichen Entscheidungen im einseitigen Verfahren statt-finden<sup>160</sup>.

## 2.6 Technizität

Nach fachmännischem Verständnis ist jede Softwareentwicklung **technisch**<sup>161</sup>! Besonders gilt dies für die eigentliche Codierung<sup>162</sup>. Das Ausschlusskriterium für Programme für Datenverarbeitungsanlagen<sup>163</sup> verkommt unter dem Gesichtspunkt der Technizität zur Bedeutungslosigkeit<sup>164</sup>. Für den Soft-wareentwickler gibt es keinen **technischen** Unterschied zwischen der Entwicklung eines Dispositions-, Buchhaltungs- oder Inkas-soprogramms und der Entwicklung eines Logikverifikations- oder Tippfehlerkorrek-turprogramms, auch wenn sich die System- und Softwareanforderungen und die An-wendungsfelder grundlegend unterscheiden. Technisch anspruchsvoller sind dabei häufig gerade Datenbank-Anwendungsprogramme für geschäftliche Tätigkeiten<sup>165</sup>.

Eine Einschränkung des Patentrechts nur auf Lehren auf den klassischen Technikgebieten hätte der Gesetzgeber sehr einfach formulieren können<sup>166</sup>; die Verfasser des EPÜ haben diese in

<sup>158</sup>BGH I ZR 52/83 a.a.O.

<sup>159</sup>BGH I ZR 139/89 a.a.O.

<sup>160</sup>vgl. hierzu *Kahin* „The Expansion of the Patent Sys-tem: Politics and Political Economy“ *firstmonday* 2001 [http://www.firstmonday.org/issues/issue6\\_1/kahin/index.html](http://www.firstmonday.org/issues/issue6_1/kahin/index.html)

<sup>161</sup>*Whittaker Atkin* „What is Enough“ *IEEE Software* 9/10 2002 10 „We firmly stand behind our belief that software de-velopment is fundamentally a technical problem and that software processes must center around technology and dis-cipline.“; *Balzert* „Was ist Software-Technik?“ in „Lehrbuch der Software-Technik“ Band 1, 2000, Seiten 35-41

<sup>162</sup>vgl. T 0049/99 „Information model based on a physi-cal system“ a.a.O.; *Hunt, Thomas* „Ubiquitous automation“ a.a.O.

<sup>163</sup>PatG § 1 Abs 2 3. i.V.m. Abs 3

<sup>164</sup>*Klopmeier* „Software und softwarebezogene Erfindun-gen — Der Schutz von Innovationen im Spannungsfeld von Urheberrecht und Patentrecht“ Seite 48

<sup>165</sup>*Balzert* a.a.O. Seite 29-30: „Die Entwicklung von Sys-temen zur Unterstützung von Vertrieb, Service und Logis-tik erfordert einen Aufwand von bis zu 100 Personenjahren und ist durch Standardsoftware nicht abgedeckt“

<sup>166</sup>*Kiesewetter-Köbinger* „Über die Patentprüfung von Pro-grammen für Datenverarbeitungsanlagen“ *GRUR* 3/2001, 185-193

„Rote Taube“<sup>167</sup> gefällte Entscheidung aber be-wußt nicht als Teil der Definition patentierbarer Gegenstände aufgenommen<sup>168</sup>.

An technischem Charakter mangelt es ei-ner Softwareanmeldung allenfalls, wenn led-iglich der naheliegende Wunsch offenbart wird, menschliche Gedankentätigkeit — z.B. die Berechnung einer bekannten physika-lisch/technisch/mathematischen Formel an-stelle mit Bleistift und Papier, einem Rechen-schieber oder einem Taschenrechner — auf den Computer zu portieren<sup>169</sup>. Die tatsächli-che Realisierung (Codierung) muss sich aber mit allen technischen Randbedingungen des Computers, seiner physikalischen Zahlendar-stellung, seinem begrenzten Wertebereich und seinem begrenzten Befehlssatz auseinander-setzen<sup>170</sup>. Dies gilt gleichermaßen für Buch-haltungsprogramme und Spiele<sup>171</sup>, wenn sie, wie gewünscht funktionieren sollen. Die Be-urteilung, dass gerade der Programmcode als (nicht-technisches) „Computerprogramm als solches“<sup>172</sup> anstatt als technische Problemlö-sung<sup>173</sup> angesehen werden soll, ist für den Softwareentwickler vom fachmännischem Ge-sichtspunkt nicht nachvollziehbar und er-scheint daher willkürlich.

## 3 Verfassungsrechtliche Be-denken

Vom Gesetzgeber eng gesteckte Schranken des urheberrechtlich geschützten Eigentums an Computerprogrammen patentrechtlich erwei-tern zu wollen, geht sicherlich über „die analoge Anwendung **einfachgesetzlicher** Vorschrif-ten“<sup>174</sup> hinaus. Anstatt ein konkurrierendes Rechtssystem aufzubauen, sollte man zunächst fragen, wieso sich urheberrechtlich der Schutz-bereich nach dem Willen des Gesetzgebers und

<sup>167</sup>BGH X ZB 15/67 GRUR 12/1969 672-674

<sup>168</sup>*McCarthy* „\*\*I Entwurf eines Berichts über den Vorschlag für eine Richtlinie des Europäischen Parla-ments und des Rates über die Patentierbarkeit com-puterimplementierter Erfindungen“ KOM(2002)92 - 2002/0047(COD), 13. Feb. 2003 Begründung 6. <http://www.europarl.eu.int/meetdocs/committees/juri/20030219/488980de.pdf>

<sup>169</sup>*Klopmeier* a.a.O.

<sup>170</sup>vgl. *Giese* a.a.O. und *Fetzer* a.a.O.

<sup>171</sup>Gegenteiliger Meinung ist *Klopmeier* a.a.O., auch wenn – wie bekannt – gerade Computerspiele die höchsten Anforderungen an Geschwindigkeit, Speicherausbau und Graphikfähigkeit fordern.

<sup>172</sup>*Tauchert* a.a.O. (siehe Fußn. 101)

<sup>173</sup>*Klopmeier* a.a.O.

<sup>174</sup>BVerfG 1 BvR 1186/89 a.a.O. (siehe Fußn. 81); BVerfG 1 BvR 479/92, 1 BvR 307/94 a.a.O. (siehe Fußn. 80)

nach juristischer Gesetzesauslegung nicht auf die gewünschte Funktionalität von Computerprogrammen erstreckt<sup>175</sup> und ggf. das dafür zuständige Urheberrecht ändern. Ob bei der derzeitigen patentrechtlichen Praxis, mit den drei Leitentscheidungen „Sprachanalyse-einrichtung“, „Logikverifikation“ und „Suche fehlerhafter Zeichenketten“ bereits „eindeutige Entscheidungen des Gesetzgebers durch eigene rechtspolitische Vorstellungen“ verändert werden<sup>176</sup>, ist klärungsbedürftig.

In diesem Zusammenhang ist auch klärungsbedürftig, ob die derzeitige Auslegung des Patentrechts (einfachen Rechts) hinsichtlich „technischer“ Computerprogramme durch die dafür zuständigen Gerichte, die Schwelle eines Verstoßes gegen die aus GG Art 14 Abs 1 S 1 iVm Art 20 Abs 2, 3 ergebenden Grenzen richterlicher Entscheidungsbefugnis erreicht<sup>177</sup>, wenn dabei der Verlust des urheberrechtlichen Verfügungs- und Verwertungsrechtes<sup>178</sup> durch entgegenstehenden (breiteren) patentrechtlichen Schutz auf urheberrechtlich freie Ideen, Grundsätze und Schnittstellen erzielt wird.

TRIPS Art 27 Abs 1 besagt zwar, dass Patente auf allen Gebieten der Technik erhältlich sein sollen, jedoch legt es für (fertige) Computerprogramme in Art 10 Abs 1 explizit urheberrechtlichen Schutz fest, der nach Art 13 nur in be-

<sup>175</sup>Ein Studium der von Davis („War of the Words: Intellectual Property Laws and Standardisation“ a.a.O.) aufgeführten Fälle wäre sicher auch patentrechtlich lohnend.

<sup>176</sup>BVerfG 1 BvR 1186/89 a.a.O. (siehe Fußn. 81); BVerfG 2 BvR 206/88 a.a.O. (siehe Fußn. 81)

<sup>177</sup>BVerfG 1 BvR 1143/90 a.a.O. (siehe Fußn. 80)

<sup>178</sup>BVerfG 1 BvR 765/66, MDR 1972, 23: „Es ist Sache des Gesetzgebers, im Rahmen der inhaltlichen Ausprägung des Urheberrechts sachgerechte Maßstäbe festzulegen, die eine der Natur und sozialen Bedeutung des Urheberrechts entsprechende Nutzung und angemessene Verwertung sicherstellen (GG Art 14 Abs 1 S 2).“;

BVerfG 1 BvR 352/71 GRUR 1980, 44-49: „Das Bundesverfassungsgericht hat in BVerfGE 31, 229 (238ff) entschieden, daß das vom Urheber geschaffene Werk und die darin verkörperte geistige Leistung in vermögensrechtlicher Hinsicht Eigentum im Sinne des GG Art 14 Abs 1 S 1 ist. Aus seiner verfassungsrechtlichen Gewährleistung erwächst dem Urheber die Befugnis, dieses „geistige Eigentum“ wirtschaftlich zu nutzen. Verfassungsrechtlich geschützt sind nicht nur die im Urheberrechtsgesetz einzeln normierten Vermögensrechte, sondern das potentielle Verfügungsrecht und Verwertungsrecht. Der Gesetzgeber ist im Rahmen des Regelungsauftrags nach GG Art 14 Abs 1 S 2 grundsätzlich verpflichtet, das vermögenswerte Ergebnis der schöpferischen Leistung dem Urheber zuzuordnen und ihm die Freiheit einzuräumen, in eigener Verantwortung darüber verfügen zu können. Im einzelnen obliegt ihm jedoch die Aufgabe, bei der inhaltlichen Ausprägung des Urheberrechts sachgerechte Maßstäbe festzulegen, die eine der Natur und sozialen Bedeutung des Rechts entsprechende Nutzung und angemessene Verwertung sicherstellen.“

stimmten Sonderfällen beschränkt und ausgenommen werden darf, welche „weder die normale Verwertung des Werkes beeinträchtigen noch die berechtigten Interessen des Urhebers unzumutbar verletzen“. Nachvollziehbar wären derartige patentrechtlichen Beschränkungen der urheberrechtlichen Verwertungsrechte allenfalls, wenn die gemeinwohlfördernde Leistung der Urheber von Computerprogrammen gegenüber jener der Ideengeber vernachlässigbar und folglich nicht schützenswert wäre. Im Einzelfall mag dies gelegentlich vorkommen. Ob dies dann aber allein durch die Patentbehörden und -gerichte festzustellen ist, bleibt die Frage.

Ob die bislang unterbliebene Vorlage an den Großen Senat zur Klärung, welche Computerprogramme — unter dem Gesichtspunkt ihrer wirtschaftlichen Verwertung — patentrechtliches Eigentum und welche urheberrechtliches Eigentum sind, bereits eine willkürliche, weil auf sachfremden Überlegungen beruhende Verletzung des GG Art 101 Abs 1 S 2 darstellt<sup>179</sup>, wenn die betroffene Fachwelt die Unterscheidung in technische und nichttechnische Computerprogramme weder sachlich, noch im Gesetzestext nachvollziehen kann, bei der Begriffsbestimmung von Computerprogrammen nicht mehr vorrangig auf das Verständnis von Computerfachleuten zurückgegriffen wird<sup>180</sup>, patentrechtliche Grundsätze zur vollständigen Offenbarung<sup>181</sup> ignoriert werden und der Effektivität des Patentschutzes<sup>182</sup> Vorrang vor dem Gemeinwohl eingeräumt wird, ist gleichfalls klärungsbedürftig.

Wegen der Harmonisierung durch das EPÜ und der Kollision mit den EG-Richtlinien zum Schutz von Computerprogrammen wäre auch

<sup>179</sup>BVerfG 1 BvR 1606/88, juris Web: „Der Grundsatz des gesetzlichen Richters ist dann verletzt, wenn ein Senat eines obersten Bundesgerichts die vor einer Abweichung von einer Entscheidung eines anderen Senats oder des Großen Senats gesetzlich geforderte Vorlage an den Großen Senat dieses Gerichts willkürlich unterläßt; eine nur rechtsirrtümliche Unterlassung oder eine fehlerhafte Anwendung einfachen Rechts (hier: FGO § 11 Abs 3) allein begründet noch keinen Verfassungsverstoß (vgl BVerfG, 9. Juni 1971, 2 BvR 225/69, BVerfGE 31, 145 <172>; zur Willkür vgl BVerfG, 4. Juni 1985, 1 BvR 1222/82, BVerfGE 70, 93 <97> mwN).“;

BVerfG 1 BvR 1113/86, MDR 1987, 904: „Das Recht auf den gesetzlichen Richter ist verletzt, wenn ein Gericht die Verpflichtung zur Vorlage an ein anderes Gericht (hier: Vorlagepflicht eines LG an OLG wegen Abweichung von einem Rechtsentscheid eines anderen OLG betreffend die Wirksamkeit eines Mieterhöhungsverlangens, Vergleiche MietRÄndG 3 Art 3 Abs 1 S 1), willkürlich außer acht läßt (Vergleiche BVerfG, 29. Juni 1976, 2 BvR 948/75, BVerfGE 42, 237, st Rspr).“

<sup>180</sup>BGH X ZB 16/00 a.a.O.

<sup>181</sup>z.B. in Form von Quellcode

<sup>182</sup>BGH X ZB 11/98 a.a.O.

an eine Vorlagepflicht beim EuGH zu denken<sup>183</sup>, wie dies auch vom EU-Parlament vorgeschlagen wurde<sup>184</sup>. Widersprechende EG-Richtlinien<sup>185</sup> sind im Falle des Eigentumsrechtes an Computerprogrammen die Richtlinie 91/250/EWG und die in der Diskussion befindliche Richtlinie 2002/0047(COD) über die Patentierbarkeit computerimplementierter Erfindungen<sup>186</sup>.

Als Patentprüfer solcher Prüfungsgebiete mit Softwareanmeldungen steht man vor dem Dilemma, entweder den Vorgaben des BGH trotz verfassungsrechtlicher Bedenken zu folgen<sup>187</sup>, oder seinem Amtseid<sup>188</sup>.

<sup>183</sup>BVerfG 1 BvR 1036/99, juris Web: „Es ist geklärt, dass der Europäische Gerichtshof gesetzlicher Richter im Sinne des GG Art 101 Abs 1 S 2 ist und es einen Entzug des gesetzlichen Richters darstellt, wenn ein nationales Gericht seiner Pflicht zur Anrufung des Europäischen Gerichtshofs im Wege des Vorabentscheidungsverfahrens nicht nachkommt (vgl. BVerfGE 73, 339 <366 ff.>; 82, 159 <194 ff.>); s. auch BVerfG, Beschluss der 1. Kammer des Ersten Senats vom 21. August 1996 - 1 BvR 866/96 -, NVwZ 1997, S. 481; BVerfG, Beschluss der 2. Kammer des Ersten Senats vom 5. August 1998 - 1 BvR 264/98 -, DB 1998, S. 1919). Danach wird die Vorlagepflicht insbesondere in solchen Fällen unhaltbar gehandhabt, in denen ein letztinstanzliches Gericht seine Vorlageverpflichtung grundsätzlich verkennt. ...“;

BVerfG 2 BvR 197/83 NJW 1987, 577-582 Leitsatz 2: „Solange die Europäischen Gemeinschaften, insbesondere die Rechtsprechung des Gerichtshofs der Gemeinschaften einen wirksamen Schutz der Grundrechte gegenüber der Hoheitsgewalt der Gemeinschaften generell gewährleisten, der dem vom Grundgesetz als unabdingbar gebotenen Grundrechtsschutz im wesentlichen gleichzuachten ist, zumal den Wesensgehalt der Grundrechte generell verbürgt, wird das Bundesverfassungsgericht seine Gerichtsbarkeit über die Anwendbarkeit von abgeleitetem Gemeinschaftsrecht, das als Rechtsgrundlage für ein Verhalten deutscher Gerichte oder Behörden im Hoheitsbereich der Bundesrepublik Deutschland in Anspruch genommen wird, nicht mehr ausüben und dieses Recht mithin nicht mehr am Maßstab der Grundrechte des Grundgesetzes überprüfen; entsprechende Vorlagen nach GG Art 100 Abs 1 sind somit unzulässig.“

<sup>184</sup>McCarthy „Einzelstaatliche Gerichte können und einzelstaatliche letztinstanzliche Gerichte müssen bei Zweifeln über die Auslegung dieser Richtlinie den Gerichtshof anrufen, wodurch ebenfalls Rechtssicherheit gewährleistet wird.“ a.a.O. Erwägung 5 (siehe Fußn. 168)

<sup>185</sup>analog zu BVerfG 1 BvR 1036/99

<sup>186</sup>[http://europa.eu.int/eur-lex/de/com/pdf/2002/de\\_502PC0092.pdf](http://europa.eu.int/eur-lex/de/com/pdf/2002/de_502PC0092.pdf), <http://www.europarl.eu.int/meetdocs/committees/juri/20030324/490455de.pdf>, <http://www.europarl.eu.int/meetdocs/committees/juri/20030219/488980de.pdf>, <http://www.europarl.eu.int/meetdocs/committees/juri/20030219/487019de.pdf>

<sup>187</sup>also der Rechtsprechung (Recht) nach GG Art 20 Abs 3 bzw nach einem Beschwerdeverfahren PatG § 79 Abs 3 S 2

<sup>188</sup>also dem vorrangigen Gesetz nach GG Art 20 Abs 3 iVm BBG § 58 Abs 2 und § 56 Abs 1

## 4 Schlussbemerkung

Bei der Arbeit an diesem Artikel kam ich wieder einmal zu der Erkenntnis, dass es wichtig ist, bei Fachartikeln nicht nur den Artikel selbst<sup>189</sup>, sondern im Vergleich hierzu auch die dort zitierte Literatur<sup>190</sup> vergleichend zu Rate zu ziehen. Dies möchte ich daher meinen Lesern ebenfalls ans Herz legen, damit sie überprüfen können, ob ich nur Humbug<sup>191</sup> erzähle. Die meisten der genannten computerwissenschaftlichen Artikel sind auch für Laien verständlich und für die Patentprüfung wertvoll. Gelegentlich ist es auch ganz nützlich, sich über die Person des jeweiligen Autors und seine Qualifikation zum Thema ein Bild zu machen<sup>192</sup>.

Für die wertvollen Anmerkungen zu diesem Artikel gilt mein besonderer Dank *Prof. N. Wirth, G. Schölch* und *A. Barthel*.

<sup>189</sup>von Hellfeld a.a.O. (siehe Fußn. 9)

<sup>190</sup>Wirth N. „Datenstrukturen und Algorithmen“ a.a.O.

<sup>191</sup>lesenswert: Frankfurt „On Bullshit“ In: „The Importance of what we care about Philosophical Essays“ Cambridge University Press <http://www.jelks.nu/misc/articles/bs.html>

<sup>192</sup>Der unbewiesene These aus den Anfängen der KI-Forschung, dass Maschinen menschengleich denken, welche z.B. Bammé vertrat und der von Hellfeld begeistert folgt, widersprechen u.a. Weizenbaum („Computermacht und Gesellschaft“ Suhrkamp 2001), Fetzer („Computers and Cognition: Why Minds are not Machines“ Kluwer 2001), Holmes („The Great Term Robbery“ IEEE Computer Mai 2001 94-96), Setzer („People are not Machines“ Pragmatics and Cognition 2/2001, 293-312), Searle („Is the Brain a Digital Computer?“ <http://www.ecs.soton.ac.uk/~harnard/Papers/Py104/searle.comp.html>) und Zemanek („Claude E. Shannon (1916-2001)“ it+ti 4/2001 177-183) aufs heftigste.

Auch zur Beurteilung, was von einem normalen Softwareentwickler erwartet werden kann, ziehe ich als Quelle E.W. Dijkstra, D.L. Parnas, N. Wirth, D.E. Knuth, F.P. Brooks, S. McConnell, R.L. Glass, N. Holmes, H. Balzert und H. Zemanek vor.