

## A. Zielsetzung

Die vorliegende Arbeit hat zum Ziel, die häufig sehr zersplittert geführte Diskussion rund um den Schutz und die Verwertung von Software im deutschen Recht einer kohärenten Gesamtbetrachtung zu unterziehen. Sie soll den Schutzrechtsgegenstand Software sozusagen „von der Wiege bis zur Bahre“ begleiten und die sich jeweils stellenden Einzelfragen in einen größeren Kontext einordnen.

Zunächst setzt sich diese Arbeit mit der terminologischen Einordnung von Software auseinander und gibt einen kurzen Einblick in die verschiedenen technischen Aspekte von Computerprogrammen. Anschließend wird auf die Notwendigkeit des (rechtlichen) Schutzes von Computersoftware sowie dessen historische Entwicklung eingegangen.

Der Hauptteil der Arbeit beginnt mit einer Untersuchung der potenziellen Schutzrechte und ihrer Einschlägigkeit für Computerprogramme. Darauffolgend wird die äußerst umstrittene Frage, ob Software nach deutschem Recht Sachqualität besitzt oder nicht, näher beleuchtet. Aus der Antwort hierauf leitet sich die Folgefrage rund um die vertragstypologische Zuordnung von Softwareverträgen ab. Hierbei werden Softwareverträge in mehrere Untergruppen aufgeteilt und für jede dieser Untergruppen jeweils die passende Vertragsart dargestellt. Der Fokus dieses Abschnitts liegt dabei neben der bloßen Anwendbarkeit des jeweiligen Vertragsinstituts auch auf den sich daraus ergebenden Leistungspflichten der Vertragsparteien sowie den Gewährleistungsrechten des Abnehmers. Ein besonderes Augenmerk liegt hierbei auf der temporären Überlassung von Software in Form eines Lizenzvertrages.

Diese werden im Anschluss an die Untersuchung der rechtlichen Verwertungsoptionen auch hinsichtlich ihrer Insolvenzfestigkeit näher beleuchtet. Hierbei wird besonders auf verschiedene Strategien zur Schaffung einer Insolvenzfestigkeit vertieft eingegangen.

Im jeweiligen Kontext werden außerdem der Schutz sowie die Verwertung von Know-how als eigenständiger Vertragsgegenstand aufgegriffen und parallel zum Software-schutz dargestellt.

Abschließend untersucht diese Arbeit den kürzlich vorgelegten Gesetzesentwurf zum Verbraucherschutz für Verträge über digitale Inhalte und zeigt auf, an welchen Stellen sich dieser Entwurf in die bisher bestehende Rechtspraxis einfügen bzw. an welchen Stellen der Gesetzesentwurf zu einem Bruch mit dieser führen würde.

Mit der Beantwortung dieser Fragen und Problemstellungen möchte diese Arbeit einen Beitrag dazu leisten, den Schutz sowie die Verwertung von Software und Know-

how stärker als einheitliches Rechtsgebiet zu begreifen und gleichzeitig Lösungsvorschläge unterbreiten, die über einzelne Teilrechtsgebiete hinweg zu einer befriedigenden Gesamtlösung führen.

## B. Terminologische Einordnung von Computersoftware

### I. Einleitung

Computersoftware wird in der Informatik als Sammelbegriff für die Gesamtheit der Programme, der zugehörigen Daten und der notwendigen Dokumentation, die es erlauben mit Hilfe eines Computers Aufgaben zu erledigen, definiert.<sup>1</sup> Weiterhin sieht der Duden Software als das Komplementär zu Hardware.<sup>2</sup>

Eine Definition von Software enthält auch die ISO-Norm ISO/IEC/IEEE 24765:2017. Diese Norm, welche die inzwischen veraltete DIN-Norm 44300 ersetzt, bietet drei mögliche Definitionen für Software an:

1. Alle Teile eines Programms und die dazugehörige Dokumentation, die dazu dienen, ein Datenverarbeitungssystem zu betreiben.
2. Software als Programm sowie die dazugehörige Dokumentation, die dazu dienen, ein Computersystem zu betreiben.
3. Programme oder eine Gruppe von Programmen, die dazu dienen, einen Computer zu betreiben.

Welche Definition einschlägig ist, hänge vom jeweiligen Kontext ab.<sup>3</sup> Diese sehr unscharfe Begriffsbestimmung, die das Vorliegen von Software zusätzlich von konkreten äußeren Bedingungen abhängig macht, ist nicht hilfreich, um eine rechtlich allgemeingültige und kohärente Definition von Software zu finden.

Auch ein Blick in das BGB führt hier zu keinem Erkenntnisgewinn. Zwar kennt das BGB seit der Schuldrechtsreform 2002 den Begriff der Software in § 312g II Nr. 5 BGB<sup>4</sup>; allerdings ist dieser nicht legal definiert. In diesem Zusammenhang wird auch in der Literatur kritisiert, dass im Rahmen der Umsetzung der Richtlinie über den Rechtsschutz von Computerprogrammen<sup>5</sup> kein Versuch unternommen wurde, eine allgemeingültige Definition von Software vorzunehmen.<sup>6</sup>

Ein Teil der Schwierigkeit einer überzeugenden Definition lässt sich mit der immateriellen Natur von Software begründen, die es erfahrungsgemäß immer erschwert, ei-

---

<sup>1</sup> *Lassmann*, Wirtschaftsinformatik, S. 127; *Horn*, Informatik, S. 311.

<sup>2</sup> Duden, Begriffsdefinition Software, abzurufen unter:  
<https://www.duden.de/rechtschreibung/Software>

<sup>3</sup> ISO/IEC-Norm 24765:2017

<sup>4</sup> Bis 2014: § 312d IV Nr. 2 BGB

<sup>5</sup> Richtlinie 2009/24/EG des Europäischen Parlaments und des Rates vom 23. April 2009 über den Rechtsschutz von Computerprogrammen.

<sup>6</sup> *Marly*, GRUR 2012, 773, 773 ff.

ne Begriffsbestimmung vorzunehmen.<sup>7</sup> Es bietet sich daher an, zunächst das körperliche Gegenstück zu Software zu betrachten und sich anschließend im Rahmen einer Negativdefinition der Bestimmung des Softwarebegriffs anzunähern.

## II. Hardware

Unter Hardware versteht man die physischen Komponenten einer Datenverarbeitungs- bzw. Computeranlage. Traditionell wird Hardware nach der Von-Neumann-Architektur<sup>8</sup> in vier Unterkategorien eingeteilt: Steuerwerk, Rechenwerk, Speicherwerk und Ein- bzw. Ausgabegerät. Auch wenn diese Einteilung aus der Mitte des zwanzigsten Jahrhunderts stammt und heute durch die Vermischung einzelner Kategorien nicht in jedem Fall trennscharf unterschieden werden kann, ist sie trotzdem hilfreich, um zu verdeutlichen, welche Komponenten zur Hardware zählen und wie sie zur Aufgabenerfüllung zusammenwirken. Auf dieser Ansicht zu Hardware aufbauend, hat auch die EG-Kommission in ihrer Entscheidung vom 24.03.2004, COMP/C-3/37.792 Microsoft ausgeführt, dass ein Computersystem aus Hard- und Software besteht, wobei Hardware die physischen Komponenten eines Computersystems bilden.<sup>9</sup> Daraus ergibt sich, dass Hardware lediglich die Grundvoraussetzung für eine funktionierende Datenverarbeitungsanlage darstellt, die erst durch Computerprogrammen in der Lage ist, konkrete Probleme zu bearbeiten und einer Lösung zuzuführen.<sup>10</sup>

Es kann also zunächst festgestellt werden, dass Software in Form von Computerprogrammen verkörpert ist.

## III. Software

Ähnlich wie Hardware, lassen sich Computerprogramme in mehrere Untergruppen einteilen<sup>11</sup>: So gibt es Systemprogramme (bspw. das Betriebssystem), Unterstützungsprogramme (bspw. Virenschanner) und Anwendungsprogramme (bspw. ein Computerspiel oder eine Warenbestellsoftware). Eine solche, auf dem Problemlösungsansatz aufbauende Definition von Computerprogrammen, findet sich auch in den Mustervorschriften der Weltorganisation für geistiges Eigentum (WIPO) von 1977. Computerprogramme werden hier definiert als „eine Folge von Befehlen, die

---

<sup>7</sup> Siehe hierzu auch Definition Know-how, S. 47 ff.

<sup>8</sup> Benannt nach dem österreichisch-ungarischen Mathematiker John von Neumann.

<sup>9</sup> EG-Kommission, 24.03.2004, COMP/C-3/37.792 Microsoft, Rn. 21 (Im englischen Originalwortlaut: Computer systems are constituted of hardware and software. The word hardware refers to the set of physical components that can constitute computer systems (such as e.g. a display, a keyboard, a hard disk or a processor).

<sup>10</sup> So auch: *Marty*, Praxishandbuch Softwarerecht, Rn. 5.

<sup>11</sup> ISO/IEC 2382:2015.

*nach Aufnahme in einen maschinenlesbaren Träger fähig sind, zu bewirken, dass eine Maschine mit informationsverarbeitenden Fähigkeiten eine bestimmte Funktion oder Aufgabe oder ein bestimmtes Ergebnis anzeigt, ausführt oder erzielt.*"<sup>12</sup> Ebenso zählt die Mustervorschrift als unter den Begriff Software fallende Programmbeschreibungen<sup>13</sup> eines Computerprogramms und das Begleitmaterial<sup>14</sup>. Sie ist damit ähnlich umfassend wie die ISO-Norm ISO/IEC/IEEE 24765:2017 und deren diverse Vorgänger. In Ermangelung einer anderen Definition hat sich diese sowohl in der Literatur<sup>15</sup> als auch in der Rechtsprechung<sup>16</sup> durchgesetzt.<sup>17</sup>

Anders als die WIPO hat sich sowohl der europäische als auch der deutsche Gesetzgeber wiederholt dagegen entschieden eine Softwaredefinition vorzunehmen.<sup>18</sup> Hintergrund dürfte die Vermutung sein, dass durch technische Fortentwicklung ein Softwarebegriff schnell überholt sein könnte.<sup>19</sup> Wie auf den vorangegangenen Seiten dargestellt, haben sich seit der Mitte des letzten Jahrhunderts der grundsätzliche Aufbau und die Funktionsweise eines Computers nicht signifikant verändert. Das Argument des Gesetzgebers scheint daher verfehlt, sodass es wünschenswert wäre, wenn der Gesetzgeber -insbesondere im Hinblick auf die Rechtssicherheit - eine Legaldefinition einführen würde. Solange dies nicht geschieht, werden sich Literatur und Gerichte weiter an der (sehr weiten) und kritikwürdigen Definition der WIPO orientieren.<sup>20</sup>

Hauptkritikpunkt der Softwaredefinition der WIPO ist, dass auch Begleitmaterial und Programmbeschreibungen erfasst sein sollen. Es erscheint hochgradig fraglich, warum dies gerade bei Software der Fall sein sollte; insbesondere da dies an anderer Stelle für Hardware explizit verneint wird.<sup>21</sup> Eine Aufnahme von Begleitmaterial in den Schutzbereich von Software könnte sich demzufolge höchstens aus der besonderen (und nicht anders zu gewährenden) Schutzwürdigkeit des Begleitmaterials ergeben.

---

<sup>12</sup> Mustervorschriften für den Schutz von Computersoftware, GRUR Int. 1978, 286; § 1 (i).

<sup>13</sup> Ebda § 1 (ii).

<sup>14</sup> Ebda § 1 (iii).

<sup>15</sup> So bspw.: Dreier/Schulze UrhG/Dreier, §69a Rn. 12, *Nebel/Stiermerling*, CR 2016, 61, 62 f.

<sup>16</sup> So etwa zuletzt: LG Hamburg, Urteil vom 03.05.2016, 408 O 46/16, MMR 2016, 782, 783.

<sup>17</sup> *Marly*, Praxishandbuch Softwarerecht, Rn. 8 ff.

<sup>18</sup> So etwa im Rahmen des Vorschlags für die Richtlinie des Rates der EG über den Rechtsschutz von Computerprogrammen, ABL EG Nr. C 91 vom 12.04.1989 oder der Begründung des Regierungsentwurfs BT-Drucks. 12/4022, S. 9.

<sup>19</sup> Dieser Ansicht zustimmend: *Loewenheim*, Handbuch Urheberrecht, § 9 Rn. 45; ablehnend hingegen: *Marly*, Praxishandbuch Softwarerecht, Fn. 25 zu Rn. 11.

<sup>20</sup> So etwa unter explizitem Verweis auf die nicht vorgenommene Legaldefinition durch den Gesetzgeber: OLG Frankfurt, Urteil vom 22.03.2005 – Az. 11 U 64/04, GRUR-RR 2005, 299, 300 (hier machte die fehlende Legaldefinition sogar einen das Urteil prägenden Streitentscheid erforderlich).

<sup>21</sup> DIN 44300 Teil 1 Nr. 1.12.

Unter Begleitmaterial versteht man in der Regel ein verschriftlichtes Dokument, das Anweisungen zur richtigen Nutzung des Programmes enthält (bspw. ein Handbuch). Ein solches Dokument ist bereits durch § 2 I Nr. 1 UrhG als Schriftwerk geschützt. Gleiches gilt auch für Programmbeschreibungen, sofern es sich bei ihnen um ein ausformuliertes Werk handelt. Sollte die Programmbeschreibung hingegen in Zeichnungen oder Skizzen verkörpert sein, so würde sie unter den Schutz des § 2 I Nr. 7 UrhG fallen.<sup>22</sup>

Es besteht daher keine Notwendigkeit, die Begleitmaterialien unter den Schutzbereich des Computerprogramms zu subsumieren. Eine gegenteilige Auffassung ließe sich allenfalls damit begründen, dass die Anforderungen für das Erlangen eines Urheberrechts an einem Computerprogramm niedriger sind als die zur Erlangung eines sonstigen Urheberrechts.<sup>23</sup> Es erscheint jedoch nicht überzeugend, dass Begleitmaterialien, die im Grundsatz nichts anderes als sonstige Schriftwerke oder technische Zeichnungen darstellen, in den Genuss des gleichen Privilegs kommen sollen. Dies lässt sich auch damit begründen, dass die Absenkung der Anforderungen zur Erlangung eines Urheberrechts bei Computerprogrammen vor allem aus verfahrens- bzw. beweistechnischen Überlegungen heraus erfolgte.

Ein weiteres Abgrenzungsproblem stellen sogenannte reine Daten, daher Sammlungen von Datensätzen dar. Mit Softwareprogrammen haben sie zunächst gemein, dass beide nicht physisch sind und ggf. auf dem gleichen Medium (Datenträger) verkörpert werden. Im Unterschied zu Computerprogrammen haben sie jedoch keine eigene Befehlsfunktion, sondern stellen lediglich eine Informationsquelle für das Computerprogramm dar. Aufgrund dieser Unterscheidung der Funktionalität sind Datensätze nicht mit Computersoftware gleichzusetzen und daher nicht unter den Begriff Computerprogramme zu subsumieren.<sup>24</sup> Diese Unterscheidung kann anhand eines gerichtlich entschiedenen Beispiels verdeutlicht werden: Während ein Computerspiel ein geschütztes Computerprogramm darstellt<sup>25</sup>, sind die gespeicherten Spielstände als bloße Informationsquelle, um den aktuellen Spielfortschritt aufzurufen, lediglich als Datensatz zu qualifizieren.<sup>26</sup> Etwas anders ausgedrückt handelt es sich bei einem Spielstand um eine für sich nicht lauffähige Ansammlung von Daten, die lediglich in Verbindung mit dem zugrundeliegenden Spiel, welches auch ohne den Spielstand gestartet werden kann, nutzbar sind.

---

<sup>22</sup> Dreier/Schulze UrhG/Dreier, § 69a Rn. 15; Marly, Praxishandbuch Softwarerecht, Rn. 12 f.

<sup>23</sup> Dreier/Schulze UrhG/Schulze, § 2 Rn. 127.

<sup>24</sup> Dreier/Schulze UrhG/Dreier, § 69a Rn. 16; Marly, Praxishandbuch Softwarerecht, Rn. 25 ff.

<sup>25</sup> Sofern es in den Schutz eines hierfür einschlägigen Gesetzes, wie etwa dem UrhG fällt.

<sup>26</sup> OLG Hamburg, Urteil vom 12.03.1998 – Az. 3 U 226/97, NJW-RR 1999, 483.

In der Rechtsprechung ist diese Abgrenzung zwischen Computerprogrammen und reinen Daten jedoch nicht unumstritten. So qualifizierte ebenfalls das OLG Hamburg in einem Fall die .ini-Datei eines Computerspiels als eigenständiges Computerprogramm.<sup>27</sup> Diese vom Gericht vertretene Ansicht überzeugt jedoch nicht. Eine .ini-Datei enthält lediglich eine Ansammlung von Werten, auf die das Computerprogramm nach bzw. bei seiner Ausführung zurückgreift. So ist bspw. in einer solchen Datei festgelegt, mit welcher Bildschirmauflösung ein Programm bei seinem Start ausgeführt wird. Die Datei selbst enthält daher keinerlei Befehle und ist auch selbst nicht in der Lage, eigenständig Rechen- oder Lösungsoperationen durchzuführen<sup>28</sup>. Sie als eigenständiges Computerprogramm zu werten, widerspricht daher dem weitverbreiteten und oben dargestellten Ansatz auf die Ausführung von Befehlen abzustellen. Folgte man der Ansicht des Gerichtes, so würde die Definition von Computerprogrammen extrem weit ausgelegt und diverse Datentypen, die nach dem Verständnis der herrschenden Meinung keine Computerprogramme sind, würden in den Schutzbereich selbiger aufgenommen werden. Im Unterschied zu einem Spielstand stellt eine .ini-Datei aber einen unverzichtbaren Teil eines Computerprogrammes dar, ohne den das Programm nicht lauffähig ist. Das gleiche Ergebnis hätte das Gericht erreichen können, wenn es die .ini-Datei nicht als eigenständiges Computerprogramm, sondern als (einen) Teil des Computerspiels, das unstrittig als Computerprogramm geschützt ist, qualifiziert hätte.

#### **IV. Zusammenfassung**

Insgesamt lässt sich feststellen, dass der Begriff Software eng auszulegen ist. Er umfasst nicht das Begleitmaterial oder die Programmbeschreibung. Ebenso wenig sind reine Datensätze erfasst. Software lässt sich daher definieren als Computerprogramm, das durch Befolgung von festgelegten Befehlen unter Nutzung zur Verfügung gestellter Informationen auf einem physischen System Aufgaben bearbeitet oder bestimmte Funktionen ausführt. Der im weiteren Verlauf dieser Arbeit genutzte Software-Begriff orientiert sich daher an der hier gefundenen Definition. Sofern andere (Neben-)Aspekte eines Computerprogrammes, wie etwa das Begleitmaterial, ebenfalls thematisiert werden, so werden diese ausdrücklich benannt.

---

<sup>27</sup> OLG Hamburg, Urteil vom 12.03.1998 – Az. 3 U 228/97, CR 1999, 298.

<sup>28</sup> Das Gericht selbst erkennt hier wohl die genaue Funktion einer .ini-Datei, in dem es feststellt, dass die .ini-Datei Befehle beinhalte. Dies ist bei einer .ini-Datei jedoch, wie oben dargestellt, nicht der Fall.

## **V. Unterschiedliche Erscheinungsformen von Computerprogrammen**

Der Kern einer Software sind immer die in ihr enthaltenen Befehle, mit deren Hilfe Probleme gelöst oder Aufgaben ausgeführt werden. Diese Befehle sind immer in einer sog. Programmiersprache verfasst und können in zwei Arten von Erscheinungsformen unterteilt werden:

### **1. Maschinenorientierte Programmiersprache**

Die maschinenorientierte Programmiersprache stellt die wahrscheinlich verbreitetste Programmiersprache dar. Sie besteht lediglich aus 0 und 1 und wird daher auch als binäre Befehlssprache bezeichnet. Befehle in Binärsprache lassen sich immer nur auf dem Prozessor umsetzen, auf dessen technischer Architektur die Befehle aufbauen. Strukturell bestehen diese Befehle immer mindestens aus einem Operations- und einem Adressteil.<sup>29</sup>

Aufgrund der starken Unübersichtlichkeit und der Fehleranfälligkeit eines so aufgebauten Befehls, werden binäre Befehle häufig in einer anderen Darstellungsform visualisiert. So ist es etwa üblich, den Operations- oder Adressteil entweder als Symbol oder als Dezimalzahl darzustellen. In der Literatur werden Befehle, die in dieser Form dargestellt sind, als Befehle in Symbol- bzw. Assemblersprache bezeichnet. Zur Ausführung eines in Symbol- bzw. Assemblersprache verfassten Befehls ist es daher nötig, dass dieser zunächst mit Hilfe eines Übersetzungsprogrammes in Binärsprache umgewandelt wird.<sup>30</sup>

### **2. Problemorientierte Programmiersprache**

Von der maschinenorientierten ist die problemorientierte Programmiersprache abzugrenzen. In dieser Sprache verfasste Befehle basieren in ihrem Aufbau nicht auf der Architektur des jeweils verwendeten Prozessors, sondern orientieren sich an der konkreten Aufgabenstellung. Sie basiert allerdings nicht auf 0 und 1 wie die Binärsprache, sondern mathematischen Formeln oder der menschlichen Sprache. Sie kann daher als Weiterentwicklung der Symbol- bzw. Assemblersprache verstanden werden. Die Vorteile einer solchen Programmiersprache (die teilweise auch als höhere Programmiersprache bezeichnet wird) liegen auf der Hand: So ist diese Sprache deutlich zugänglicher, was es auch Personen, die mit dem jeweiligen Programm nicht vertraut sind, nach kurzer Einarbeitungszeit ermöglicht, die Befehle zu bearbeiten. Auch die Fehlersuche ist bei solchen Programmen deutlich einfacher. Ähnlich wie bei in Symbol- bzw. Assemblersprache verfassten Befehlen, müssen auch solche in höherer Pro-

<sup>29</sup> Dworatschek, Grundlagen der Datenverarbeitung, S. 338 f.

<sup>30</sup> Prinz/Crawford, C in a nutshell, S. 495.



grammiersprache zunächst in binäre Befehle umgewandelt werden, um auf dem jeweiligen Prozessor lauffähig zu sein. Dies geschieht entweder durch einen Kompilierer oder einem Interpretierer. Ein Kompilierer wandelt das gesamte Programm in höhere Programmiersprache gleichzeitig um, während ein Interpretierer immer nur einzelne Befehle „auf Abruf“ umwandelt.<sup>31</sup>

Sind Befehle nicht in Binärsprache verfasst, sondern müssen erst umgewandelt werden, so spricht man von Quellprogrammen.<sup>32</sup> Ohne Zwischenschritte ausführbare Programme werden hingegen als Objektprogramme bezeichnet.<sup>33</sup>

---

<sup>31</sup> *Marly*, Praxishandbuch Softwarerecht, Rn. 22 f.

<sup>32</sup> *König*, NJW 1992, 1731, 1731.

<sup>33</sup> *König*, GRUR 1989, 559, 560 f.

## **C. Notwendigkeit des rechtlichen Schutzes von Computersoftware**

Nachdem nun eine allgemein anwendbare Definition von Computersoftware herausgearbeitet wurde, ist damit noch nicht die Frage beantwortet, ob es auch notwendig ist, Software rechtlich zu schützen. Gleiches gilt für die Frage, wie ein rechtlicher Schutz konkret ausgestaltet sein könnte und welches Rechtsinstitut hierfür hauptsächlich in Frage kommt. Um sich dieser Frage weiter zu nähern ist es sinnvoll, sich zunächst mit den grundsätzlichen Erwägungen hinsichtlich des Schutzes von immateriellem Vermögen auseinanderzusetzen. Anschließend sollen die einzelnen Stakeholder einer Computersoftware differenziert betrachtet werden.

### **I. Grundsätzliche Erwägungen bzgl. des Schutzes immateriellen Vermögens**

Die grundsätzliche Unterscheidung zwischen materiellem und immateriellem Vermögen liegt in der offensichtlichen Divergenz, dass immaterielles Vermögen nicht in einer physischen Form existiert. Dies gilt ausdrücklich auch für Computersoftware.<sup>34</sup> Selbige mag zwar häufig auf einem Datenträger gespeichert sein, allerdings ist dieser Datenträger lediglich das Transportmedium der Computersoftware.<sup>35</sup> Diese Differenzierung lässt sich anhand eines kurzen Beispiels verdeutlichen: Die Frage, ob ein beschriebenes Stück Papier eine Sache iSd. § 90 BGB darstellt, wird lediglich mit Blick auf das Vorliegen der Sacheigenschaften des Schriftstücks beurteilt. Ob sich das Schriftstück temporär oder dauerhaft in einem Briefumschlag befindet, spielt bei der Beurteilung dieser Frage keine Rolle.

Aus der nicht-physischen Form ergibt sich, dass immaterielles Vermögen an mehreren Orten gleichzeitig existieren kann. Im Gegensatz zu einer physisch vorhandenen Sache ist ein immaterieller Gegenstand daher theoretisch beliebig oft reproduzierbar und unterliegt damit keiner natürlichen Nutzungsbegrenzung. Es ließe sich dementsprechend argumentieren, dass ein Schutz von geistigem Eigentum gar nicht nötig sei. Es stehe schließlich unbegrenzt zur Verfügung und könne von allen gleichermaßen und gleichzeitig genutzt werden.<sup>36</sup>

Dem gegenüber steht die Erwägung, dass die Schaffung geistigen Eigentums (daher die Erfindung) genauso auf einem Schöpfungsprozess fußt wie die Erstellung eines

---

<sup>34</sup> Kraßer/Ann PatenR/Ann, § 2, Rn. 1.

<sup>35</sup> Hierzu detailliert: S. 58 ff.

<sup>36</sup> Besonders durch das Erstarken der Piratenpartei(en) auf bundesdeutscher sowie europäischer Ebene hat dieses Argument wieder an Befürwortern gewonnen und eine gewisse Renaissance erfahren. Es bleibt abzuwarten, ob sich dieses Argument auch nach dem Ausscheiden der Piratenpartei(en) aus div. Parlamenten im politischen Diskurs nachhaltig etabliert.