

---

# **Der Sourcecode von Computerprogrammen im schweizerischen Recht und in der EU-Richtlinie über den Rechtsschutz von Computerprogrammen**

VON RA DR. WOLFGANG STRAUB, Bern\*

## **Inhaltsübersicht**

### I. Der Sourcecode als Schlüsselement der Softwareentwicklung

1. Vom Sourcecode zum Objectcode
2. Interessen des Erwerbers am Sourcecode
3. Interessen des Herstellers am Sourcecode

### II. Die Rechte am Sourcecode

1. Schutzobjekt
2. Recht auf Erstveröffentlichung
3. Kopier- und Nutzungsrecht
4. Programmänderungen
5. insbesondere Fehlerbehebung
6. Ansprüche auf Herausgabe des Sourcecodes
7. Analyse des Objectcodes
8. Rekonstruktion des Sourcecodes

### III. Escrow und Hinterlegung

### IV. Ergebnisse

\* Für wertvolle Hinweise danke ich den Herren Dr. Stephan Beutler, LL.M., Dr. Beda Bischof, Dr. Ralph Schlosser, LL.M., und Herrn dipl.-Ing. Hansjörg Wälchli.

## I. Der Sourcecode als Schlüsselement der Softwareentwicklung

### 1. Vom Sourcecode zum Objectcode

Computer können Programme nur in digitalisierter Form «verstehen», d.h. als Abfolge von binären Zahlen. Die Programmierung wäre in dieser Form impraktikabel, besteht doch jeder einzelne Befehl an die Hardware aus einer kaum überschaubaren Reihe von Einsen und Nullen. Computerprogramme werden daher in *höheren Programmiersprachen*<sup>1</sup> geschrieben, welche eine logische Verknüpfung von Befehlsworten erlauben und sozusagen eine eigene Grammatik zur Verfügung stellen, in welcher die Programminhalte formuliert werden können. Es braucht jedoch weitere Arbeitsschritte, um solche Programme für den Computer ausführbar zu machen.

Computerprogramme durchlaufen verschiedene Entwicklungsstadien:

- Zuerst wird in der Regel ein *Konzept* erarbeitet, welches den strukturellen Aufbau des Programms festlegt. Es bildet zusammen mit allfälligen weiteren relevanten Informationen über Schnittstellen etc. die *Entwicklungsdokumentation*<sup>2</sup>.
- Das Konzept wird realisiert, indem Befehl für Befehl, Zeile für Zeile in eine Programmiersprache umgesetzt (implementiert) und Verweise auf andere Unterprogramme und Programmbibliotheken eingebaut werden. Das Resultat ist der *Sourcecode*.

<sup>1</sup> In der Informatik wird allgemein zwischen *maschinennahen Programmiersprachen* (insbesondere Assemblersprachen) und *höheren Programmiersprachen* (z.B. Basic, C, Pascal, Modula2, Java etc.) unterschieden. Erstere sind der Arbeitsweise der Hardware soweit angepasst, dass ihre Umsetzung in den binären Maschinencode (*executable*) relativ einfach ist. Sie erlauben daher, Anweisungen an die Hardware sehr präzise zu formulieren und hohe Ausführungsgeschwindigkeiten zu erzielen, sind jedoch nur relativ unbequem zu programmieren. Wegen ihrer rationelleren und übersichtlicheren, dem menschlichen Denken leichter zugänglichen Ausdrucksweise werden die meisten kommerziellen Programme in höheren Programmiersprachen geschrieben, welche auf die entsprechende Aufgabenstellung zugeschnitten sind.

<sup>2</sup> Zusätzliche Informationen zum Programmaufbau werden allerdings oft nicht in einer separaten Entwicklungsdokumentation erfasst, sondern nur als Bemerkungen in den Sourcecode hineingeschrieben.

- Mit *Compiler*<sup>3</sup> und *Linkerprogrammen*<sup>4</sup> werden die Befehle der Programmiersprache später in Anweisungen umgesetzt, welche in den Arbeitsspeicher geladen werden können. Die durch die Kompilierung hergestellte Form wird nachfolgend als *Objectcode*<sup>5</sup> bezeichnet.

Das fertige Computerprogramm erzeugt zusammen mit dem Betriebssystem eine *Benutzeroberfläche* (z.B. Windows-Dialogboxen), über die der Anwender mit dem Computerprogramm kommunizieren kann.

## 2. Interessen des Erwerbers am Sourcecode

Für die Benutzung eines Programms auf einem Computer ist grundsätzlich nur der Objectcode erforderlich. Der Erwerber kann dennoch ein großes Interesse an der Mitlieferung des Sourcecodes haben: Spätere *Änderungen*<sup>6</sup> (Fehlerbehebung, Weiterentwicklung etc.) sind praktisch oft nur durch Bearbeitung des Sourcecodes möglich. Auch die Einbindung eines Computerprogramms in eine IT-Umgebung, ins-

<sup>3</sup> Statt der einmaligen Umsetzung des ganzen Sourcecodes in Objectcode werden z.T. *Interpreterprogramme* verwendet, die sozusagen eine Simultanübersetzung des Sourcecodeablaufs herstellen, was allerdings die Ausführungsgeschwindigkeit reduziert. Die Programmiersprache *Java* basiert auf diesem Prinzip.

<sup>4</sup> Ein Computerprogramm besteht in der Regel aus verschiedensten Unterprogrammen, welche nicht notwendigerweise alle in der Form des Sourcecodes vorliegen müssen. Sind die Schnittstellen bekannt, kann auch ein in Objectcodeform vorliegender Programmbaustein in das Gesamtprogramm eingebunden werden. Durch das Linkerprogramm werden verschiedene in Objectcode umgewandelte Programmteile mit einander und mit Programmbibliotheken (DLL) verknüpft und sozusagen in einen Rahmen eingebettet, welcher das anschließende Laden des ganzen Programms in den Arbeitsspeicher erlaubt. *Compiling* und *linking* müssen für den Programmierer allerdings nicht zwei sichtbar getrennte Schritte darstellen.

<sup>5</sup> In der Informatik wird unterschieden zwischen *Objectcode* (Objekte, d.h. kompilierte Programme bzw. Programmbausteine, welche für die Hardware noch nicht direkt verständlich, für Menschen aber praktisch nicht mehr lesbar sind) und *Binärcode* (in binäre Information umgewandelte und mit einander verknüpfte Objekte, auch als *executables* bezeichnet). Entsprechend der in der juristischen Literatur inzwischen allgemein üblichen Terminologie wird der Begriff des Objectcodes nachfolgend indessen auch für den Binärcode verwendet.

<sup>6</sup> Vgl. zur Zulässigkeit von Programmänderungen weiter hinten II.4.

besondere die Datenkommunikation mit anderen Softwarekomponenten setzt gewisse Informationen über Schnittstellen des Computerprogramms (*Interfaces*) voraus, welche im Sourcecode enthalten sind. Die Erwerber von teuren und individuell entwickelten Computerprogrammen haben zudem ein erhebliches Interesse am Sourcecode, wenn nicht sicher ist, ob der Hersteller zur längerfristigen *Weiterentwicklung* und Anpassung technisch und wirtschaftlich in der Lage und zu angemessenen Konditionen bereit sein wird.

### 3. *Interessen des Herstellers am Sourcecode*

Die Offenlegung des Sourcecodes ermöglicht grundsätzlich, das betreffende Computerprogramm abzuändern<sup>7</sup> oder Teile davon in andere Programme zu übernehmen, was mit urheberrechtlichen Mitteln nur teilweise verhindert werden kann.

Größere Computerprogramme bestehen in der Regel sowohl aus urheberrechtlich schützbaaren Teilen, als auch aus Sequenzen, welche zwar Aufwand und Know-how<sup>8</sup> beinhalten können, die Schwelle urheberrechtlichen Schutzes jedoch nicht erreichen<sup>9</sup>. Indessen ist oft schwierig zu beurteilen, wie weit der urheberrechtliche Schutz reicht, sodass erhebliche Unsicherheit darüber besteht, welche Teile eines Sourcecodes von Dritten verwendet werden dürfen.

<sup>7</sup> Da einheitliche Regeln für den Aufbau von Computerprogrammen weitgehend fehlen, ist der oft aus hunderttausenden von Programmzeilen bestehende Sourcecode für sich alleine allerdings auch für Spezialisten manchmal so unübersichtlich, dass eine effiziente Weiterentwicklung des Programms ergänzende Informationen erfordert. Auch allfällig vorhandene Entwicklungsdokumentationen sind allerdings von höchst unterschiedlicher Qualität bzw. inhaltlicher Dichte, sodass in gewissen Fällen der Bezug der ursprünglichen Programmierer zur Weiterentwicklung unverzichtbar bleibt. In anderen Fällen ist der Sourcecode sozusagen «selbsterklärend» und mit allen nötigen Hinweisen zu Aufbau und Funktion der einzelnen Programmteile versehen, sodass er eine Weiterentwicklung ohne weiteres erlaubt.

<sup>8</sup> Vgl. zum *Know-how-Schutz* von Computerprogrammen *Ralph Schlosser, Le contrat de savoir-faire*, Diss., Lausanne 1996, S. 45 ff.

<sup>9</sup> Vgl. zum urheberrechtlichen Schutz von Computerprogrammen nachfolgend II.1.

Die Offenlegung des Sourcecodes erhöht nicht nur die Gefahr von Urheberrechtsverletzungen<sup>10</sup>. Sie führt für den Inhaber des Urheberrechts<sup>11</sup> zudem zu verschiedenen *wirtschaftlichen Risiken*: Konkurrenten erhalten durch die Übernahme von urheberrechtsfreien Programmteilen und im Sourcecode enthaltenem Know-how Wettbewerbsvorteile. Wenn der Erwerber des Programms durch den Sourcecode in die Lage versetzt wird, urheberrechtlich zulässige Anpassungen und Weiterentwicklungen<sup>12</sup> selbst vorzunehmen, können Geschäftsmöglichkeiten verloren gehen, sofern nicht bereits lang dauernde Wartungsverträge mit Pauschalvergütung bestehen.

## II. Die Rechte am Sourcecode

Nachfolgend soll ein Überblick über die besonderen Aspekte des urheberrechtlichen Schutzes von Sourcecodes gegeben werden. Urheberrechtliche Befugnisse wie z.B. das Recht auf Urhebernennung, welche kaum spezifische Probleme in Bezug auf den Sourcecode stellen, bleiben daher unberücksichtigt<sup>13</sup>.

### 1. Schutzobjekt

*Urheberrechtlich schützbar* können sowohl Sequenzen einzelner Befehle als auch der strukturelle Aufbau<sup>14</sup> eines Computerprogramms

<sup>10</sup> So können z.B. Seriennummern, Herstellername und Urheberrechtsvermerke anhand des Sourcecodes einfach entfernt und direkte Raubkopien hergestellt werden. Vgl. dazu den in SMI 1994, S. 181 ff. publizierten Entscheid des Obergerichts Zürich vom 13.11.1993 (Netware *serial updating program*). Wer den Sourcecode kennt, kann ihn zudem soweit umzustellen versuchen, dass die Feststellung einer urheberrechtsverletzenden Übereinstimmung mit dem ursprünglichen Programm erheblich erschwert wird.

<sup>11</sup> Unter *Inhaber des Urheberrechts* werden auch Personen verstanden, welche Urheberrechte vom eigentlichen Urheber derivativ erworben haben (vgl. dazu insbesondere Art. 17 URG).

<sup>12</sup> Vgl. zur urheberrechtlichen Zulässigkeit der Änderung von Computerprogrammen nachfolgend II.4.

<sup>13</sup> Vgl. zum Urheberrecht an Computerprogrammen insbesondere die systematische Darstellung von *Neff/Arn* in SIWR II/2.

<sup>14</sup> Vgl. zum Schutz der *Programmstruktur* insbesondere *Botschaft*, BBl. 1989 III 523, sowie *Neff/Arn*, SIWR II/2, S. 141. Zwischen schützbarer Programmstruktur

sein, nicht aber die Aufgabenstellung<sup>15</sup> und die dem Programm zugrundeliegenden Ideen<sup>16</sup>. Entscheidend ist, ob die Elemente eines Computerprogramms einzeln oder zusammen *individuell* sind: Obwohl Computerprogramme als urheberrechtsfähige Werke in Art. 2 Abs. 3 URG<sup>17</sup> separat erwähnt ist, geht die schweizerische Lehre davon aus, dass auch sie nur dann urheberrechtlich schützbar sind, wenn sie den allgemeinen Anforderungen von Art. 2 Abs. 1 URG entsprechen, d.h. geistige Schöpfungen mit individuellem Charakter darstellen<sup>18</sup>.

tur und den darin verkörperten nicht schützbaeren Ideen besteht indessen ein Spannungsverhältnis. Vgl. dazu nachfolgend Fn. 16.

<sup>15</sup> Computerprogramme stellen Lösungen einer bestimmten *Aufgabenstellung* dar, welche selbst nicht urheberrechtlich schützbar ist. Vgl. dazu *Wolfgang Straub*, Individualität als Schlüsselkriterium des Urheberrechts, GRUR Int. 2001, S. 1 ff., S. 5.

<sup>16</sup> Nach der *Botschaft*, BBl. 1989 III, S. 523, fallen die einem Programm zugrunde liegenden Lösungsprinzipien in der Schweiz nicht unter das Urheberrecht. Inwieweit die urheberrechtlich schützbaere Form eines Werkes gegenüber nicht schutzfähigen *Ideen* abgegrenzt werden kann und soll, ist in der Schweiz indessen schon seit BGE 19, S. 943 ff., kontrovers. Vgl. dazu grundlegend *Ivan Cherpillod*, L'objet du droit d'auteur, étude critique de la distinction entre forme et idée, Diss. Lausanne 1985, S. 59 ff. und 171 ff. Vgl. auch *Kamen Troller*, Manuel du droit suisse des biens immatériels, 2. A., Basel/Frankfurt a.M. 1996, Band I, S. 244 f., und Band II, S. 313 ff., sowie *François Dessemontet*, Le droit d'auteur, Lausanne 1999, Rz. 141 ff.

Art. 1 Abs. 2 CRRL (Fn. 19) hält entsprechend dem 14. Erwägungsgrund ausdrücklich fest, dass Ideen und Grundsätze, die irgendeinem Element eines Computerprogramms zugrunde liegen nicht urheberrechtlich geschützt seien. Vgl. dazu *Jochen Marly*, Urheberrechtsschutz für Computersoftware in der Europäischen Union, Abschied vom überkommenen Urheberrechtsverständnis, München 1995, S. 120 ff. und 128 ff., *Bridget Czarnota/Robert J. Hart*, Legal Protection of Computer Programs in Europe – A Guide to the EC Directive, London/Dublin/Edinburgh/München 1991, S. 36, sowie rechtsvergleichend *Alain Strowel*, Droit d'auteur et copyright, divergences et convergences, étude de droit comparé, Paris 1993, S. 413 ff.

<sup>17</sup> Art. 2 Abs. 3 des durch das IGE ausgearbeiteten Vorentwurfs zur Revision des URG (nachfolgend VE URG) entsprechend Art. 10 Absatz 1 TRIPS bzw. Art. 4 WIPO copyright Treaty präzisiert, dass Computerprogramme ungeachtet der Art oder Form des Ausdrucks als *Sprachwerke* geschützt sind.

<sup>18</sup> *Individualität* setzt nach der hier vertretenen Auffassung sowohl einen Gestaltungsspielraum, als auch dessen individuelle Ausnützung voraus, sodass ein statistisch gesehen einmaliges Resultat entsteht. Es muss mit andern Worten praktisch ausgeschlossen werden können, dass ein Dritter unabhängig auf die selbe Lösung kommen würde. Vgl. dazu *Neff/Arn*, SIWR II/2, S. 138 ff., sowie *Max Kummer*, Das urheberrechtlich schützbaere Werk, Bern 1968, S. 43 ff., und *Straub* (Fn. 15), S. 3 ff., mit weiteren

Aus der Entstehungsgeschichte des schweizerischen URG geht hervor, dass hinsichtlich der Bestimmungen über Computerprogramme Konformität mit der *EU-Richtlinie über den Rechtsschutz von Computerprogrammen* (nachfolgend CRRL)<sup>19</sup> angestrebt wurde<sup>20</sup>. Art. 1 Abs. 3 CRRL scheint die Kriterien der Individualität und der geistigen Schöpfung einander gleichzusetzen. Ob daraus allerdings der Schluss gezogen werden kann, dass in der EU an Computerprogramme geringere Anforderungen als an sonstige Werke gestellt werden müssen, ist fraglich<sup>21</sup>. Auf jeden Fall kann für sie nicht eine höhere Individualität als für andere Werkkategorien verlangt werden.

Literaturnachweisen. Ein Schutz von nicht individuellen Programmen bzw. Sequenzen, widerspricht den Prinzipien des Urheberrechts, welches – im Gegensatz zu Art. 5 lit. c UWG – nicht auf den Schutz des wirtschaftlichen Aufwands abstellt, sondern nur die Monopolisierung von Leistungen mit individuellem Charakter ermöglicht. Vgl. dazu allerdings auch *Denis Barrelet/Willi Egloff*, Das neue Urheberrecht, Kommentar zum Bundesgesetz über das Urheberrecht und verwandte Schutzrechte, 2. A. Bern 2000, N. 25 zu Art. 2 URG.

<sup>19</sup> Richtlinie vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen, publiziert in ABl. L. 122/42 vom 17.5.1991 (Modifikationen in ABl. L. 290/9 vom 24.11.1993 und ABl. L. 001/482 vom 3.1.1994). Vgl. zur *Entstehungsgeschichte der CRRL* *Czarnota/Hart* (Fn. 16), S. 7 ff., *Marly* (Fn. 16), S. 311 ff., *Jerôme Huet*, L'Europe des logiciels: les droits des utilisateurs, Recueil Dalloz, partie chronique, 1992, S. 315 ff., *Michael Lehmann*, Die Europäische Richtlinie über den Schutz von Computerprogrammen, GRUR Int. 1991, S. 327 ff. Die Richtlinie ist in den EU-Staaten ins Urheberrecht transponiert worden, sodass ihre Bestimmungen via das nationale Recht angewendet werden.

<sup>20</sup> Vgl. dazu *Felix H. Thomann*, Die Euroverträglichkeit der Softwareschutz-Regelung gemäss dem revidierten URG, AJP 1993, S. 563 ff., S. 568.

<sup>21</sup> Vgl. dazu *Marly* (Fn. 16), S. 104 ff., 114 ff. und 123 ff., wonach der CRRL ein sich am angelsächsischen Recht orientierender *europäischer Schöpfungsbegriff* zugrunde liegt, welcher auf dem Einsatz von Können und Mühe beruht. Vgl. dazu auch *Czarnota/Hart* (Fn. 16), S. 43 f. Vgl. zur Entwicklung der angloamerikanischen *sweat of the brow doctrine* *Strowel* (Fn. 16), S. 443 ff. und 461 ff., und *Jean-Sylvestre Bergé*, La protection internationale et communautaire du droit d'auteur, essai d'une analyse conflictuelle, Paris 1996, S. 22 f. Der U.S. Supreme Court hat sich von diesem Ansatz in der Entscheidung *Feist Publications Inc. vs. Rural Tel. Service Co.*, 499 U.S. 340 (1991), inzwischen allerdings distanziert. Ein am Aufwand orientierter Ansatz klingt übrigens teilweise auch in der Rechtsprechung des schweizerischen Bundesgerichts an, wenn es auf einen geistigen Mindestaufwand abstellt. Vgl. z.B. BGE 100 II 167 E. 7.

Urheberrechtlichen Schutz können auch die *Entwicklungsdokumentation*<sup>22</sup> und die *Benutzeroberfläche*<sup>23</sup> – bzw. Teile davon – genießen, falls sie selbst ausreichende Individualität aufweisen.

Bei der *Kompilierung* durchläuft der Sourcecode einen Transformierungsprozess, welcher nach den im betreffenden Compilerprogramm festgelegten Regeln erfolgt. Immerhin können einzelne Sourcecodebefehle auf unterschiedliche Weise in Objectcode umgesetzt werden. Für alle wichtigen Programmiersprachen sind unterschiedliche Compilerprogramme erhältlich, welche auf bestimmte Aufgabenstellungen zugeschnitten sind und dementsprechend je zu einem etwas anderen Objectcode führen. Die Transformierung selbst ist indessen ein rein maschineller Prozess und keine geistige Schöpfung<sup>24</sup>. Am Objectcode besteht somit kein über den Sourcecode hinausgehender Urheberrechtsschutz.

Hingegen enthält der Sourcecode in der Regel Informationen, welche für den Rechenprozess irrelevant sind, aber für das Verständnis und die Weiterentwicklung des Programms von Bedeutung sind (z.B. Parameternamen, Bemerkungen des Programmierers zum Aufbau des Programms, zu Schnittstellen etc.). Diese Informationen, welche ausnahmsweise eigene Individualität aufweisen können, werden nicht in Objectcode umgewandelt. Der *Sourcecode* kann aufgrund dieser zu-

<sup>22</sup> Vgl. zum weit gefassten Schutzobjekt von Art. 1 Abs. 2 CRRL *Czarnota/Hart* (Fn. 16), S. 33 f. Auch in Hardwarebauteilen verkörperte Software (*Firmware*) fällt unter die CRRL.

<sup>23</sup> Ob die *Benutzeroberfläche* selbst urheberrechtlichen Schutz genießt, hängt von ihrer Individualität ab. Vgl. zur Schutzfähigkeit von Programmoberflächen nach schweizerischen Recht *Neff/Arn*, SIWR II/2, S. 150 f. Vgl. zum Verhältnis zwischen Sourcecode und Benutzeroberfläche nach der CRRL *Marly* (Fn. 16), S. 141 ff. Die Benutzeroberfläche wird letztlich durch den Sourcecode bestimmt. Nach der hier vertretenen Konzeption kann urheberrechtlicher Schutz für verschiedene Ebenen des selben Werkes bestehen (z.B. Befehlssequenz, Metastruktur, Benutzeroberfläche). Sowohl die vom Betriebssystem zur Verfügung gestellten Module (Dialogfelder etc.) als auch die Art und Weise, wie diese mit einander bzw. mit der Struktur des Anwendungsprogramms verknüpft werden, können daher selbständig urheberrechtlich schutzbar sein.

<sup>24</sup> Vgl. dazu auch *Helmut Haberstumpf*, Der urheberrechtliche Schutz, in: Michael Lehmann (Hrsg.), Rechtsschutz und Verwertung von Computerprogrammen, 2. A., Köln 1993, S. 148, Rz. 145.

sätzlichen Informationen gegenüber dem Objectcode unter Umständen *zusätzlichen urheberrechtlichen Schutz* genießen<sup>25</sup>.

Schließlich können technische Erfindungen, welche sich auf Computerprogramme beziehen, in der EU<sup>26</sup> und in der Schweiz *patentiert* werden<sup>27</sup>. Für das grafische Layout von Programmoberflächen kommt auch eine Hinterlegung als *Muster* bzw. *Design* in Betracht. Zudem kann das direkte Kopieren von Computerprogrammen, welche die Schwelle immaterialgüterrechtlichen Schutzes noch nicht erreichen, ausnahmsweise gegen Art. 5 lit. c UWG verstoßen<sup>28</sup>.

## 2. Recht auf Erstveröffentlichung

Der Urheber hat das alleinige Recht, über die Erstveröffentlichung seines Werks zu entscheiden<sup>29</sup>. Veröffentlichen bedeutet, ein Werk ei-

<sup>25</sup> Das Bundesgericht geht in BGE 125 III 263 davon aus, dass aus der *Änderungsberechtigung* am Objectcode nicht ohne weiteres auf entsprechende Rechte am Sourcecode geschlossen werden kann. Hingegen dürften aus Rechten am Sourcecode in der Regel entsprechende Rechte am Objectcode abgeleitet werden können. Hat der Nutzer bei einer Entwicklungslizenz ausnahmsweise nur den Sourcecode erhalten, so stellt die Herstellung eines Objectcodes daraus eine Nutzungshandlung dar, welche durch Art. 12 Abs. 2 URG abgedeckt ist.

<sup>26</sup> Art. 52 Abs. 2 lit. c EPÜ schließt die *Patentierung* ganzer Programme für Datenverarbeitungsanlagen zwar grundsätzlich aus. Softwarebezogene *technische Erfindungen* können indessen patentiert werden. Vgl. dazu den Überblick über die Praxis des EPA bei *Margarete Singer/Dieter Stauder*, Europäisches Patentübereinkommen, Kommentar, 2. A., Köln/Berlin/Bonn/München 2000, N. 36 ff. zu Art. 52 EPÜ. Mit dem Entscheid der Beschwerdekammer vom 1.7.1998, T-1173/97 (IBM Programmprodukt) wurde die bisherige Praxis stark liberalisiert. Vgl. dazu *Jürgen Betten*, European Viewpoint of IP Patentability, LES Nouvelles 2000, S. 25 ff.

<sup>27</sup> Die *Praxis des schweizerischen IGE* lehnt sich an diejenige des EPA an. Vgl. dazu Ziff. 223.4 der Richtlinien des IGE für die Sachprüfung der Patentgesuche sowie zur Entwicklung bis 1998 *Peter Heinrich*, Kommentar zum Schweizerischen Patentgesetz und den entsprechenden Bestimmungen des Europäischen Patentübereinkommens, Zürich 1998, N. 1.17 ff. zu Art. 1 PatG, *Alexandra Frei*, Softwareschutz durch das Patentrecht, in: Felix H. Thomann/Georg Rauber (Hrsg.), Softwareschutz, Bern 1998, S. 97 ff.

<sup>28</sup> Das Urheberrecht schließt die Anwendung von Bestimmungen des *Lauterkeitsrechts* auf Computerprogramme zwar nicht *a priori* aus, indessen darf immaterialgüterrechtlich versagter Schutz nicht über den Umweg des UWG «erschlichen» werden. Spezialgesetzlich erlaubtes Verhalten verstößt nach der allerdings nicht unbestrittenen bundesgerichtlichen Praxis jedenfalls nur unter besonderen Umständen gegen die Generalklausel von Art. 2 UWG. Vgl. dazu etwa BGE 104 II 322 E. 5b und 97 II 78 E. 3.

<sup>29</sup> Art. 9 Abs. 2 URG. Vgl. dazu auch Art. 4 lit. c CRRL.

ner größeren Anzahl Personen<sup>30</sup> *außerhalb eines privaten Kreises zugänglich zu machen*<sup>31</sup>. Ob es dann tatsächlich zur Kenntnis genommen wird, ist unerheblich<sup>32</sup>. Dementsprechend führt die *Publikation im Internet* generell zu einer Erstveröffentlichung<sup>33</sup>.

Die urheberpersönlichkeitsrechtliche Frage der Erstveröffentlichung hat nur relativ beschränkte praktische Bedeutung für Computerprogramme, da für das *Vervielfältigen* ohnehin Art. 10 Abs. 2 URG<sup>34</sup> vorbehalten bleibt<sup>35</sup>. Immerhin ist im Fall der *Zwangsvollstreckung* eine Verwertung von noch nicht erstveröffentlichten Werken grundsätzlich ausgeschlossen<sup>36</sup>.

Es fragt sich, ob für den *Sourcecode* unabhängig von einer allfälligen zusätzlichen Individualität gegenüber dem Objectcode ein *eigenes*

<sup>30</sup> Art. 9 Abs. 3 URG spricht von einer *größeren Anzahl Personen*. Oft beschränkt sich das mögliche Interesse an einem Programm bereits auf einen relativ kleinen Personenkreis (z.B. innerhalb einer bestimmten Branche). Ist der Sourcecode diesem Kreises zugänglich, dürfte es als erstveröffentlicht gelten, da ein weiter gehendes Veröffentlichungsinteresse fehlt.

<sup>31</sup> Art. 9 Abs. 3 URG verweist hinsichtlich des *privaten Kreises* auf Art. 19 Abs. 1 lit. a URG. Diese Norm ist nach Abs. 4 auf Computerprogramme indessen nicht direkt anwendbar. Nach *Neff/Arn*, SIWR II/2, S. 211, führt die bloße Weitergabe eines Programms an Testpersonen noch nicht zu einer Veröffentlichung. Sie stellen auf die Geheimhaltungsmöglichkeit gegenüber einem größeren Kreis ab. In BGE 96 II 409 (Gold Rush) geht das Bundesgericht davon aus, dass das Erstveröffentlichungsrecht durch die Übertragung der Urheberrechte an einem Film noch nicht erschöpft wird. Vgl. dazu auch *Kamen Troller*, Précis du droit suisse des biens immatériels, Basel/Genf/München 2001, S. 235. Ist der Sourcecode eines Programms also nicht erstveröffentlicht, wenn die Rezipienten vertraglich zur Geheimhaltung gezwungen werden können? Dies dürfte solange zu bejahen sein, als es sich um eine begrenzte Zahl von Geheimnisträgern handelt, sodass die Geheimhaltung auch faktisch noch einigermaßen kontrollierbar ist. Vgl. dazu *Barrelet/Egloff* (Fn. 17), N. 23 zu Art. 9 URG.

<sup>32</sup> Vgl. dazu etwa *Barrelet/Egloff* (Fn. 17), N. 24 zu Art. 9 URG.

<sup>33</sup> Vgl. dazu auch *Lukas Bühler*, Schweizerisches und internationales Urheberrecht im Internet, Diss. Fribourg 1999, S. 223. Immerhin fragt sich, ob eine Erstveröffentlichung im Internet auch dann stattgefunden hat, wenn ein Werk auf einer zwar theoretisch einem größeren Kreis zugänglichen Internetseite sozusagen *versteckt* (z. B. ohne direkte Aufrufmöglichkeit via Hyperlinks) publiziert wurde.

<sup>34</sup> Vgl. zur internationalen *Bedeutung der Erstveröffentlichung als Schutzvoraussetzung* nach Art. 3 Abs. 1 lit. b RBÜ und Art. II Abs. 1 WUA *Bühler* (Fn. 33), S. 222.

<sup>35</sup> Vgl. dazu nachfolgend II.3.

<sup>36</sup> Vgl. dazu *Manfred Reh binder*, Schweizerisches Urheberrecht, 3. A., Bern 2000, S. 146, Rz. 133, und *Barrelet/Egloff* (Fn. 17), N. 22 zu Art. 9 URG.

*Erstveröffentlichungsrecht* besteht<sup>37</sup>. Der Objectcode ermöglicht in der Regel nicht ohne weiteres<sup>38</sup>, die individuelle Struktur oder schutzfähige Sequenzen eines Computerprogramms zu erkennen. Nur in Ausnahmefällen erlauben die Programmoberfläche<sup>39</sup> oder sonstige mitveröffentlichte Unterlagen eine Wahrnehmung der individuellen Eigenarten des Sourcecodes. Solange er nicht veröffentlicht wurde, dürfte er somit vom Erwerber grundsätzlich nicht weitergegeben werden<sup>40</sup>. Ein Recht zur *Weitergabe* kann sich indessen aus Art. 12 Abs. 2 URG ergeben<sup>41</sup>, wonach ein vom Inhaber des Urheberrechts oder mit seiner Zustimmung in Verkehr<sup>42</sup> gebrachtes Computerprogramm grundsätzlich weiter veräußert werden darf<sup>43</sup>.

<sup>37</sup> *Neff/Arn*, SIWR II/2, S. 211, gehen hingegen davon aus, dass ein Programm als veröffentlicht gilt, wenn es als Sourcecode oder als Objectcode veröffentlicht wurde.

<sup>38</sup> Vgl. zur beschränkten Zulässigkeit der Analyse von Computerprogrammen weiter hinten II.7.

<sup>39</sup> Gleiche *Benutzeroberflächen* bedingen jedenfalls nicht zwingend identische Programmstrukturen, sodass aus der Benutzeroberfläche nicht unbedingt auf identische Sourcecodes geschlossen werden kann. Vgl. dazu *Neff/Arn*, SIWR II/2, S. 150 f., und *Marly* (Fn. 16), S. 139.

<sup>40</sup> *Dessemontet* (Fn. 16), S. 159 ff., geht für das schweizerische Recht davon aus, dass die *Veräußerung* eines Werks das Recht zur Erstveröffentlichung nicht mit einschließt, a.M. hingegen *Barrelet/Egloff* (Fn. 17), N. 21 zu Art. 9 URG. Allein die Weitergabe an einen Erwerber dürfte allerdings noch keine Erstveröffentlichung darstellen, auch wenn der Veräußerer dessen spätere Handlungen nicht kontrollieren kann. Vgl. dazu *Rehbinder* (Fn. 32), S. 146. Für den Sourcecode führt dies zur Anomalie, dass die übrigen Urheberrechte grundsätzlich schon vor einer eigentlichen Erstveröffentlichung *erschöpft* sein können.

<sup>41</sup> Wenn analog zur CRRL davon auszugehen ist, dass Art. 12 Abs. 2 URG einen *zwingenden Kerngehalt* hat (vgl. dazu auch nachfolgend II.4), ist fraglich, ob ein generelles vertragliches Weitergabeverbot für den Sourcecode im Fall der Weiterveräußerung des Programms zulässig ist.

<sup>42</sup> Ob das Programm physisch oder *online* auf den Erwerber übergegangen ist, spielt in diesem Zusammenhang keine Rolle. *Bühler* (Fn. 33), S. 263 f. und 285 f., wohl anderer Meinung hingegen *Michael Hyzik*, Kein Schutz des Softwarekonsumenten durch das Urheberrecht, Jusletter vom 19.6.2000, Rz. 10. Die Weitergabe eines Computerprogramms dürfte immerhin bei der bloßen Programmnutzung im Rahmen eines *Application Service Providing* (ASP) unzulässig sein, da hier zur bestimmungsgemäßen Nutzung gar keine dauerhafte Speicherung des Computerprogramms erfolgt. Vgl. dazu *Re to M. Hilty*, Lizenzvertragsrecht, Bern 2001, S. 263. Es fehlt in diesen Fällen generell bereits an der Tatbestandsvoraussetzung der Veräußerung des Programms.

<sup>43</sup> Art. 12 Abs. 2 URG sowie Art. 4 lit. c CRRL, welche im Gegensatz zum schweizerischen Urheberrecht allerdings von einer bloß *regionalen Erschöpfung* (EWR) aus-

Darf umgekehrt der Urheberrechtsinhaber den Sourcecode gegen den Willen der Erwerber des Programms veröffentlichen? Nach der in der Schweiz wohl herrschenden Lehre ist das Erstveröffentlichungsrecht trotz seinem persönlichkeitsrechtlichen Charakter übertragbar<sup>44</sup>, sodass auch ein *Verzicht* auf dessen Ausübung möglich ist. Aus den Vertragsumständen kann sich ergeben, dass eine Publizierung erkennbare Sicherheitsinteressen der Erwerber beeinträchtigt (z.B. Zahlungssystem einer Bank), welche eventuell von vertraglichen Nebenpflichten umfasst sind. Selbst wenn die Veröffentlichung des Sourcecodes zulässig ist, kann sich im Fall seiner Freigabe (z.B. Publikation des Sourcecodes als *Public Domain*-Programm im Internet) allenfalls ein Anspruch auf Reduktion laufender Lizenzgebühren ergeben: Soweit diese auf der Nutzung eines immateriellen Exklusivrechts beruhen, besteht eine Analogie zum Verlust der Exklusivität wegen Nichtigkeit des Immaterialgüterrechts<sup>45</sup>.

### 3. Kopier- und Nutzungsrecht

Das Vervielfältigungsrecht stellt das praktisch bedeutsamste Recht des Urheberrechtsinhabers dar<sup>46</sup>. Bereits das Laden eines Computerprogramms tangiert dieses Recht, da dabei technisch gesehen eine *vorübergehende Programmkopie* im Arbeitsspeicher des Computers entsteht<sup>47</sup>. Unter Art. 10 Abs. 2 URG dürfte auch das *Zugänglichma-*

geht. Nach BGE 124 III 321 (Nintendo) E. 1 und 2 gilt im Urheberrecht grundsätzlich die *internationale Erschöpfung*, sodass auch im Ausland rechtmäßig in Verkehr gebrachte Computerprogramme in der Schweiz weiterveräußert werden dürfen.

<sup>44</sup> Vgl. dazu bejahend *Botschaft* (Sonderdruck) S. 51., von *Büren*, SIWR II/1, S. 211, *Dessemontet* (Fn. 16), S. 150, *Neff/Arn*, SIWR II/2, S. 210, verneinend *Barrelet/Egloff* (Fn. 17), N. 19 ff. zu Art. 9 URG. Jedenfalls kann die *Ausübung der Erstveröffentlichung* übertragen werden.

<sup>45</sup> Vgl. zum Einfluss der *Nichtigkeit von Immaterialgüterrechten* auf Lizenzen von *Büren*, SIWR I/1, S. 302 ff., sowie BGE 116 II 191 E. 3.

<sup>46</sup> Der Inhaber des Urheberrechts hat grundsätzlich das alleinige Recht, Werkexemplare herzustellen (Art. 10 Abs. 2 lit. a URG/Art. 4 lit. a CRRL).

<sup>47</sup> Vgl. dazu *Neff/Arn*, SIWR II/2, S. 223 ff., *Dessemontet*, SIWR II/1, S. 174, und *Gianni Fröhlich-Bleuler*, Urheberrechtliche Nutzungsbefugnisse des EDV-Anwenders, AJP 1995, S. 569 ff., S. 573. Auch Art. 4 lit. a CRRL macht grundsätzlich jede dauerhaf-

chen im Internet fallen, da im Computer des Herunterladenden eine – wenn vielleicht auch nur vorübergehende – Werkkopie entsteht<sup>48</sup>. Allerdings ist der Erwerber<sup>49</sup> eines Computerprogramms gesetzlich ermächtigt, dieses zu gebrauchen<sup>50</sup>.

Der Gebrauch von Computerprogrammen erfolgt in der Regel lediglich in der Objectcodeform<sup>51</sup>. Soweit der Erwerber ausnahmsweise berechtigt ist, am Sourcecode Änderungen vorzunehmen<sup>52</sup> oder diesen anhand des Objectcodes zu rekonstruieren, darf er auch dazu allfällige notwendige Kopiervorgänge durchführen<sup>53</sup>.

te oder vorübergehende vollständige oder teilweise Vervielfältigung von der Zustimmung des Urheberrechtsinhabers abhängig, selbst wenn sie im Rahmen des Ladens oder Vervielfältigens erfolgt. Vgl. dazu *Michael Lehmann*, Die Richtlinie über den Schutz von Computerprogrammen, in: Michael Lehmann (Hrsg.), *Rechtsschutz und Verwertung von Computerprogrammen*, 2. A., Köln 1993, S. 12, und *Czarnota/Hart* (Fn. 16), S. 56. Art. 10 Abs. 2 lit. a VE URG will dementsprechend klarstellen, dass auch *mittelbare und vorübergehend ausdrückliche Vervielfältigungen* die Urheberrechte des Inhabers tangieren.

<sup>48</sup> Vgl. dazu *Bühler* (Fn. 33), S. 168 ff.

<sup>49</sup> Art. 12 Abs. 2 URG setzt voraus, dass das Programm «veräußert» wurde. Art. 5 Abs. 1 CRRL verwendet den Begriff des «rechtmäßigen Erwerbers» (*lawful acquirer*). Ob ein *rechtmäßiger* Erwerb stattgefunden hat, ist grundsätzlich nach dem anwendbaren nationalen Privatrecht zu beurteilen. Indessen wird keine Eigentumsübertragung an einem Werkexemplar vorausgesetzt, sodass jedenfalls auch *online* übertragene Software unter diese Bestimmung fällt. Als rechtmäßiger Erwerber kommt auch ein Lizenz- oder Leasingnehmer in Betracht. Vgl. dazu *Czarnota/Hart* (Fn. 16), S. 64 f. Vgl. ferner auch die Formulierung von Art. 21 Abs. 1 lit. b VE URG, welche die Lizenznehmer als Kategorie von Gebrauchsberechtigten erwähnt.

<sup>50</sup> Art. 12 Abs. 2 URG enthält eine *gesetzliche Lizenz* zugunsten des Erwerbers eines Computerprogramms. Vgl. zu deren Rechtsnatur weiter hinten Fn. 69. Art. 13a Abs. 1 VE URG versucht, das Gebrauchsrecht analog zu Art. 5 Abs. 1 CRRL zu präzisieren. Der Umfang der gesetzlichen Lizenz wird durch das Kriterium der *bestimmungsgemäßen Benutzung* in Abhängigkeit von den konkreten Umständen definiert. Je nachdem ob etwa eine Mehrplatznutzung beabsichtigt war, dürfen die notwendigen Kopiervorgänge für den zeitgleichen Einsatz des Programms auf einem oder mehreren Computern oder Terminals durchgeführt werden. Vgl. dazu auch *Marly* (Fn. 16), S. 198 ff.

<sup>51</sup> Vgl. zum direkten Gebrauch von Sourcecodes mit einem *Interpreterprogramm* allerdings Fn. 3. Abgesehen davon kann die *Nutzung* eines Computerprogramms in gewissen Fällen (z. B. Anpassung, Fehlerbeseitigung) den Sourcecode erfordern.

<sup>52</sup> Vgl. dazu nachfolgend II.4 und II.5.

<sup>53</sup> Vgl. dazu auch *Melchior Caduff*, Die urheberrechtlichen Konsequenzen der Veräußerung von Computerprogrammen, Bern 1997, zugl. Diss. Bern 1996, S. 101.

#### 4. Programmänderungen

Jede Modifikation eines urheberrechtlich geschützten Computerprogramms bzw. Programmteils stellt grundsätzlich eine Werkänderung dar, welche die Zustimmung des Urheberrechtsinhabers voraussetzt<sup>54</sup>.

Nicht nur die – für Computerprogramme höchstens ausnahmsweise relevanten – Ansprüche gegen Werkentstellungen<sup>55</sup>, sondern auch das allgemeine Recht auf Werkintegrität<sup>56</sup> hat seine Grundlage primär im *Persönlichkeitsrecht des Urhebers*<sup>57</sup>. Dieses wird durch die Modifikation einer beliebig wiederherstellbaren Programmkopie<sup>58</sup> nicht im gleichen Maß tangiert wie etwa durch die Veränderung eines einzigen Werkexemplars (z.B. Bauwerk). Die Interessen der Entwickler von Computerprogrammen sind primär kommerzieller Natur<sup>59</sup>. Urheberrechte werden im IT-Bereich zudem meist nicht von den eigentlichen Urhebern, sondern von ihren Arbeitgebern ausgeübt<sup>60</sup>. Wenn nur urheberpersönlichkeitsrechtliche Aspekte zu berücksichtigen sind, fragt sich daher, inwieweit überhaupt ein Rechtsschutzinteresse an der Inte-

<sup>54</sup> Art. 11 URG. Siehe auch Art. 4 lit. b CRRL und Art. 13a Abs. 1 lit. b VE URG. Vgl. dazu *Neff/Arn*, SIWR II/2, S. 214, und *Peter Hafner*, Das Verhältnis urheberrechtlicher Befugnisse zum Eigentum am Werkexemplar, Diss. Zürich 1994, Bern 1994, S. 38 f.

<sup>55</sup> Art. 11 Abs. 2 URG.

<sup>56</sup> Art. 11 Abs. 1 URG.

<sup>57</sup> Vgl. dazu *Jacques de Werra*, Le droit à l'intégrité de l'œuvre, Etude de droit d'auteur suisse dans une perspective de droit comparé, Bern 1997, zugl. Diss. Lausanne 1997, S. 41 f., *Hafner* (Fn. 54), S. 71, und *Barrelet/Egloff* (Fn. 17), N. 1 zu Art. 11 URG.

<sup>58</sup> Immerhin können durch Änderungen eines Computerprogramms neue Fehler entstehen, welche sich ausnahmsweise auf das *Image* des ganzen Programms und damit auf das Ansehen des Urhebers auswirken können.

<sup>59</sup> Die Inhaber des Urheberrechts haben insbesondere ein *wirtschaftliches Interesse*, die Programmmodifikation durch Dritte zu beschränken, um selbst entsprechende Leistungen anbieten zu können. Vgl. dazu weiter vorne I.3.

<sup>60</sup> Art. 17 URG lässt nur die Urhebervermögensrechte auf den Arbeitgeber übergehen. Indessen ist jedenfalls auch die *Übertragung der Ausübung* von Urheberpersönlichkeitsrechten zulässig. *Von Büren*, SIWR II/1, S. 210, und *Ursula Widmer*, Der urheberrechtliche Schutz von Computerprogrammen, ZSR 1993, 1. Halbband, S. 247 ff., S. 255 f., leiten aus Art. 11 Abs. 2 URG ab, dass auch eine *Übertragung des Änderungsrechts* selbst möglich ist (welche allerdings persönlichkeitsverletzende Änderungen nicht umfasst). Vgl. differenzierend *de Werra* (Fn. 57), S. 174 ff. *Dessemontet*, SIWR II/1, S. 160, nimmt bei der Übertragung von Werken mit praktischer Zielsetzung wie Computerprogrammen einen impliziten Verzicht auf die Ausübung des Rechts auf Werkintegrität an.

grität von Computerprogrammen besteht<sup>61</sup>. Die Abwehrbefugnis gegenüber Werkänderungen kann indessen auch als ein Recht mit sowohl urheberpersönlichkeitsrechtlichem als auch vermögensrechtlichem Charakter begriffen werden<sup>62</sup>.

Wenn das Änderungsrecht auch die *Vermögensinteressen* des Urheberrechtsinhabers mitschützen soll, ist *de lege lata* wohl grundsätzlich von der *Zustimmungspflicht* zu *allen Änderungen* auszugehen<sup>63</sup>.

Indessen kann sich aus dem Veräußerungs- oder Lizenzvertrag ein *ausdrückliches oder stillschweigendes*<sup>64</sup> *Modifikationsrecht* ergeben. Hat der Vertragspartner Anspruch<sup>65</sup> auf ein Exemplar des Sourcecodes, so kann daraus in der Regel auch auf eine Zustimmung zu dessen Änderung geschlossen werden<sup>66</sup>, da sein praktischer Nutzen für den Erwerber vor allem in der Modifikationsmöglichkeit liegt. Allerdings kann sich aus dem Vertragsverhältnis ergeben, dass eine Änderung nur für bestimmte Fälle (z.B. erst nach Beendigung eines parallelen Wartungsvertrages) oder nur in bestimmtem Umfang (z.B. reine Fehlerbehebung) zulässig ist oder dass der Sourcecode zu einem anderen Zweck (z.B. Herstellung von interoperablen Programmen) geliefert wurde.

Das geltende schweizerische Urheberrecht enthält im Gegensatz zur CRRL<sup>67</sup> keine explizite Modifikationsberechtigung des Erwerbers

<sup>61</sup> Vgl. zur gewandelten Bedeutung des Rechts auf *Werkintegrität für digitale Werke* auch *de Werra* (Fn. 57), S. 104.

<sup>62</sup> Vgl. dazu auch *Barrelet/Egloff* (Fn. 17), N. 1 zu Art. 11 URG.

<sup>63</sup> So auch BGE 125 III 263.

<sup>64</sup> Vgl. dazu allerdings zurückhaltend *Neff/Arn*, SIWR II/2, S. 214.

<sup>65</sup> Aus der tatsächlichen *Mitlieferung des Sourcecodes* ohne entsprechende vertragliche Grundlage kann ein Änderungsrecht nicht ohne weiteres abgeleitet werden. Vgl. dazu BGE 125 III 263 E. 4c. Immerhin können sich aus dem Verhalten nach den Vertragsschluss Indizien für den Vertragsinhalt ergeben. Vgl. dazu auch *Christine Chappuis*, Note sur le devoir d'information du donneur de licence à l'égard du preneur de licence en matière informatique, SJ 1999, S. 474 ff., S. 476.

<sup>66</sup> Vgl. dazu auch *Fröhlich-Bleuler* (Fn. 47) S. 574, und zum deutschen Recht *Andreas Günther*, Änderungsrechte des Softwarenutzers, CR 1994, S. 321 ff., S. 328.

<sup>67</sup> Art. 5 Abs. 1 in Verbindung mit Art. 4 lit. b der CRRL erlauben eine Bearbeitung von Computerprogrammen, wenn diese zur *bestimmungsgemäßen Benutzung* einschließlich der Fehlerbehebung notwendig ist. Vgl. dazu *Marly* (Fn. 16), S. 214 ff., der als Beispiele etwa die Veränderung der Tastaturbelegung, die Einbindung von Hardwaretreibern, und die Portierung auf ein anderes Computersystem erwähnt.

vom Computerprogrammen<sup>68</sup>. Trotzdem kann aus Art. 12 Abs. 2 URG<sup>69</sup> ein Änderungsrecht abgeleitet werden<sup>70</sup>, wenn der *bestimmungsgemäße Gebrauch* im Einzelfall Modifikationen mit umfasst. Im Ergebnis ist eine *Abwägung* zwischen den Interessen des Urheberrechtshabers und des Programmnutzers vorzunehmen<sup>71</sup>. Sowohl das Partizipationsinteresse des Urheberrechtshabers als auch die Änderungsinteressen des Nutzers sind letztlich rein wirtschaftlicher Natur und damit gleichartig.

Modifikationen müssen nicht von vornherein zwingend notwendig für den Programmgebrauch sein. Die bestimmungsgemäße Nutzung

<sup>68</sup> Vgl. die enge Formulierung von Art. 11 Abs. 1 und 12 Abs. 2 URG in Verbindung mit Art. 17 URV. Eine Werkänderung im Rahmen des Eigengebrauchsrechts ist für Computerprogramme sogar ausdrücklich ausgeschlossen (Art. 19 Abs. 4 URG). Hingegen besteht in Art. 12 Abs. 3 URG für Architekturwerke eine explizite Basis für Werkänderungen. Das Bundesgericht ließ die Änderung von Bauwerken gestützt auf das *Eigentumsrecht* allerdings bereits vor dem Inkrafttreten des geltenden Urheberrechtsgesetzes zu. Vgl. BGE 117 II 466 E. 4, bestätigt in BGE 120 II 65 E. 7 (ETHZ). Indessen sieht Art. 13a Abs. 1 lit. b VE URG ausdrücklich vor, dass Übersetzungen und Bearbeitungen einschließlich der Fehlerbeseitigung im Rahmen der bestimmungsgemäßen Benutzung zulässig werden sollen. Der erläuternde Bericht (S. 6) geht davon aus, dass die geltende Rechtslage durch diesen Artikel nicht berührt werde. Es wird offenbar nur eine Klärung im Sinne einer Angleichung an die CRRL beabsichtigt.

<sup>69</sup> Vgl. zur *Rechtsnatur der gesetzlichen Lizenz* von Art. 12 Abs. 2 URG *Reto M. Hilty*, Lizenzvertragsrecht (Fn. 42) S. 260 f., und ders., Der Schutz von Computerprogrammen – nationale und internationale Normen auf dem Prüfstand des Internets, sic 1997, S. 128 ff., S. 139 f. Art. 12 Abs. 2 URG konkretisiert bis zu einem gewissen Grad den Zweckübertragungsgrundsatz. Vgl. dazu analog im deutschen Recht *Haberstumpf* (Fn. 24), S. 154, Rz. 159.

<sup>70</sup> Nach *Fröhlich-Bleuler* (Fn. 47), S. 574, kann ein Änderungsrecht direkt aus Art. 2 bzw. Art. 28 ZGB abgeleitet werden. Vgl. dazu auch *Felix Christen*, Die Werkintegrität im schweizerischen Urheberrecht, Bern 1982, zugl. Diss. Zürich 1982, S. 128 ff.

<sup>71</sup> Vgl. zur *Interessenabwägung* unter dem URG *Caduff* (Fn. 53), S. 104 f., und *Fröhlich-Bleuler* (Fn. 47), S. 574. Vgl. zur CRRL *Marly* (Fn. 16), S. 220 ff. Nach *Czarnota/Hart* (Fn. 16), S. 58, fallen Änderungen, welche vom Nutzer nur deshalb selbst durchgeführt werden, um die Kosten der Wartung durch den Inhaber des Urheberrechts einzusparen, d.h. sein Partizipationsinteresse zu umgehen, von vornherein nicht unter Art. 5 CRRL. *Günther* (Fn. 66), S. 327, betrachtet ein Änderungsrecht des Nutzers dann nicht interessengerecht, wenn Anpassungsleistungen bereits in der Preiskalkulation des Urhebers erfolgten. Solche Aspekte können durchaus in die Interessenabwägung einfließen. Indessen ist zu berücksichtigen, dass die interne Preiskalkulation des Herstellers dem Softwareerwerber meist nicht zugänglich ist.

kann auch bloß *nützliche Programmanpassungen*<sup>72</sup> oder die Portierung auf eine neue IT-Umgebung<sup>73</sup> umfassen. In diesem Zusammenhang sind namentlich die *Art des Programms* und seine Aufgabenstellung von Bedeutung. Währenddem etwa der übliche Gebrauch eines Textverarbeitungsprogramms kaum Programmänderungen erwarten lässt, muss Individualsoftware welche auf sich wandelnde Betriebsabläufe und rechtliche Gegebenheiten (z.B. Steuersätze) zugeschnitten ist, fortlaufend angepasst werden.

Besteht ein Änderungsrecht, so kann mit dessen Ausführung auch ein *Dritter* beauftragt werden, da dem Erwerber selbst oft das notwendige Know-how fehlt<sup>74</sup>.

Kann ein auf Art. 12 Abs. 2 URG basierendes Änderungsrecht vertraglich *wegbedungen* werden? Dies könnte dazu führen, dass eine Software nutzlos wird, wenn der Inhaber des Urheberrechts zur Weiterentwicklung zu angemessenen Konditionen selbst nicht bereit oder in der Lage ist<sup>75</sup>. In der EU wird davon ausgegangen, dass das Ge-

<sup>72</sup> Vgl. dazu auch *Caduff* (Fn. 53), S. 104, sowie für das deutsche Recht, *Günther* (Fn. 66), S. 325 f. Da die endgültige Fassung der CRRL die *Wartung und Anpassung* nicht mehr erwähnt, ist umstritten, ob entsprechende Änderungen grundsätzlich zulässig sind. Vgl. dazu verneinend *Lehmann* (Fn. 47), S. 18, bejahend *Haberstumpf* (Fn. 24), S. 160 f.

<sup>73</sup> Vgl. zur *Portierung* von Computerprogrammen *Fröhlich-Bleuler* (Fn. 47), S. 575. IT-Systeme sind meist einem konstanten Wandel unterworfen. Um ein Computerprogramm auf einem neuen Computer, einem neuen Betriebssystem oder mit neuer Zusatzsoftware benutzen zu können, sind oft Anpassungen notwendig. Marktmächtige Hersteller von Computerprogrammen versuchen z.T. in Lizenzverträgen die *Portierung* auf andere IT-Umgebungen zu untersagen bzw. von zusätzlichen Gebühren abhängig zu machen (sog. *CPU-Klauseln*). Vgl. zur (relativen) Rechtsnatur von CPU-Klauseln *Neff/Arn*, *SIWR* II/2, S. 255, und *Marly* (Fn. 16), S. 190 ff. In diesem Zusammenhang fragt sich, ob der zu erwartende Gebrauch eines Computerprogramms auch in zeitlicher Hinsicht zu präzisieren ist. Derartige Klauseln sind jedenfalls insoweit zulässig, als sie an Stelle einer zeitlichen Limitierung der Lizenz treten und die Nutzungsdauer des IT-Systems einigermmaßen abschätzbar ist.

<sup>74</sup> Vgl. auch dazu den Entscheid des deutschen BGH vom 24.2.2000 (I ZR 141/97), publiziert in CR 2000, S. 656 ff. Dort wurde der Beizug einer Drittfirma im Rahmen einer vertraglich erlaubten Portierung als mit einer Geheimhaltungsvereinbarung vereinbar betrachtet, da die Urheberin die notwendigen Programmanpassungen nicht selbst vornehmen wollte.

<sup>75</sup> Vgl. dazu auch *Caduff* (Fn. 53), S. 103. Wenn der Urheberrechtsinhaber die notwendigen Anpassungen von Standardsoftware etwa innert angemessener Frist in neuen *Programmreleases* vornimmt, dürfte es in der Regel an einem Änderungsinteresse fehlen. Vgl. *Fröhlich-Bleuler* (Fn. 47), S. 575.

brauchsrecht zwar grundsätzlich dispositiv ist, aber einen zwingenden Kern enthält<sup>76</sup>. Eine entsprechende Auslegung des URG rechtfertigt sich auch im schweizerischen Recht<sup>77</sup>. Die Geltendmachung der urheberrechtlichen Abwehrbefugnisse kann in solchen Fällen auch mit dem *Rechtsmissbrauchsverbot* in Konflikt geraten<sup>78</sup>.

### 5. Insbesondere Fehlerbehebung

Ein besonders großes Interesse an Änderungen von Computerprogrammen besteht im Hinblick auf Fehler<sup>79</sup>. Die schweizerische Lehre ist sich praktisch einig, dass die bloße Behebung von Fehlern zulässig sein muss<sup>80</sup>. Dies wird aus einer *Abwägung* zwischen den Interessen des Urhebers an der Werkintegrität<sup>81</sup> und den Interessen des Eigentü-

<sup>76</sup> Vgl. dazu *Marly* (Fn. 16), S. 214 ff. und 229, *Czarnota/Hart* (Fn. 16), S. 65, *Günther* (Fn. 66), S. 326, und *Lehmann* (Fn. 47), S. 13 f. und 17. Vgl. auch dazu den Entscheid des deutschen BGH vom 24.2.2000 (Fn. 74), S. 656 ff., E. II/1/b/aa/3. Nach § 42d Abs. 2 bzw. 4 des österreichischen UrhG dürfen Computerprogramme zwingend vervielfältigt und bearbeitet werden, soweit dies für ihre bestimmungsgemäße Nutzung durch den zur Benutzung Berechtigten notwendig ist. Dort wird ausdrücklich festgehalten, dass dazu auch die *Anpassung an dessen Bedürfnisse* gehört.

<sup>77</sup> Vgl. dazu auch *Hyzik* (Fn. 42), Rz. 9 f.

<sup>78</sup> Vgl. dazu *Fröhlich-Bleuler* (Fn. 47), S. 576.

<sup>79</sup> Ein *Fehler im Sinn der Informatik* setzt voraus, dass ein Computerprogramm bzw. dessen Leistung von einem Sollparameter abweicht. Vgl. dazu *Heinz Bons*, Fehler und Fehlerauswertung, in: Computer-Software und Sachmängelhaftung, herausgegeben von Peter Gorny/Wolfgang Kilian, Stuttgart 1985, S. 35 ff. Ab einer gewissen Größe und Komplexität sind Fehler im Sinn der Technik praktisch unvermeidbar. Die informatikspezifische Fehlerdefinition lässt allerdings die Zweckbestimmung des Computerprogramms und die Konsequenzen der Abweichung unberücksichtigt. Fehler von Computerprogrammen sind daher grundsätzlich nur dann rechtlich relevant, wenn sie Einfluss auf deren Gebrauchstauglichkeit haben oder ein Schädigungsrisiko mit sich bringen. Vgl. dazu auch *Wolfgang Straub*, Die Haftung für das Jahr-2000-Problem, *AJP* 1999, S. 524 ff.

<sup>80</sup> Vgl. dazu *Neff/Arn*, *SIWR* II/2, S. 257, *Caduff* (Fn. 53), S. 102, und *Ivan Cherpilod* Protection des logiciels et des bases de données: la révision du droit d'auteur en Suisse, *SMI* 1993, S. 49 ff., wohl a.M. hingegen *Cyrill P. Rigamonti*, Das Jahr-2000-Computer-Problem – ein Rechtsproblem, *SJZ* 1998, S. 430 ff., S. 433.

<sup>81</sup> Statt eines urheberpersönlichkeitsrechtlich geprägten Interesses an der Werkintegrität dürfte tatsächlich eher ein *wirtschaftliches Interesse* an einer Monopolisierung der Weiterentwicklung in Betracht fallen. Vgl. dazu weiter vorne I.3.

mers<sup>82</sup> an der Nutzung des Werkexemplars abgeleitet<sup>83</sup>, was vom Ergebnis her als sachgerecht erscheint. Allerdings lässt sich eine solche Interessenabwägung nicht auf sachenrechtliche Bestimmungen abstützen<sup>84</sup>.

Der urheberrechtliche Schutz nimmt auf den Zweck eines Werkes nicht direkt Bezug<sup>85</sup>. Die Frage der *Fehlerhaftigkeit* von Computerprogrammen ist daher grundsätzlich nach den Maßstäben des *Obligationenrechts* bzw. des Produkthaftungsrechts zu beurteilen<sup>86</sup>. Urheberrechtlichen Ansprüchen zum Schutz der Werkintegrität können somit vertragliche Befugnisse<sup>87</sup> zur Mängelbeseitigung entgegen stehen.

<sup>82</sup> In Softwareüberlassungsverträgen wird dem Nutzer manchmal kein Eigentum an einem Werkexemplar, sondern nur ein *obligatorisches Nutzungsrecht* eingeräumt. Vgl. etwa den Sachverhalt in BGE 125 III 263. Diese Konstruktion soll die Lizenzkomponente (Nutzungserlaubnis) gegenüber kauf- oder werkvertraglichen Elementen stärker betonen und damit die Vertragsqualifikation beeinflussen. Unter Umständen wird auch bezweckt, die gesetzliche Lizenz des Erwerbers durch weniger weit gehende vertragliche Ansprüche zu ersetzen. Die Begriffe der Veräußerung (Art. 12 Abs. 2 URG) bzw. des rechtmäßigen Erwerbs (Art. 5 Abs. 1 CRRL) stellen allerdings nicht auf die sachenrechtliche Qualifikation, sondern auf die Nutzungsberechtigung ab (vgl. dazu Fn. 49), sodass der zwingende Kerngehalt der gesetzlichen Lizenzen (vgl. dazu weiter vorne II.4) auch bei fehlendem Eigentumsübergang am Programmexemplar gilt.

<sup>83</sup> Vgl. *Cherpillod*, SIWR II/1, S. 252, und ders. (Fn. 80), S. 61, sowie *Hafner* (Fn. 54), S. 82 ff. Vgl. in diesem Zusammenhang auch BGE 117 II 466 E. 4 und 120 II 65 E. 7 (ETHZ).

<sup>84</sup> Eigentum besteht in der Regel am Trägermaterial des Programms (Harddisk, CD-ROM, Diskette etc.). Gestützt auf Art. 641 ZGB kann dieses innerhalb der Schranken der Rechtsordnung ebenso wie jeder leere Datenträger verändert, vernichtet oder mit einer beliebigen Information überschrieben werden. Aus dem Eigentum kann wohl aber kein Recht abgeleitet werden, den Datenträger mit einem bestimmten immaterialgüterrechtlich geschützten Inhalt zu überschreiben. Hingegen besteht ein *wirtschaftliches Interesse, an der besseren Nutzung* der für den Erwerb des Programms bereits getätigten Investition.

<sup>85</sup> Das geltende *Urheberrecht* befasst sich direkt nicht mit der *Fehlerhaftigkeit* von Werken (vgl. demgegenüber Art. 13a Abs. 1 VE URG, der die Fehlerbeseitigung unter die bestimmungsgemäße Nutzung subsumiert). Das Werk ist im Sinn des Urheberrechts in der veröffentlichten Form in sich abgeschlossen (vgl. allerdings auch Art. 3 URG). Durch jede Veränderung entsteht ein vom ersten Werk meist abhängiges, eventuell seinerseits individuelles, neues Werk.

<sup>86</sup> Vgl. dazu auch *Straub* (Fn. 79), S. 426 f.

<sup>87</sup> Soweit zur Mängelbeseitigung in die Rechtssphäre des Urhebers eingegriffen werden muss, weil kein anderweitiges Fehlerbehebungsrecht besteht, ist zur *Ersatzvornahme* eine richterliche Ermächtigung nötig. Der Richter hat in diesem Fall eine Interessenabwägung vorzunehmen. Es ist kontrovers, ob im Rahmen von Art. 98 Abs. 1

Wie bei Programmmodifikationen im Allgemeinen liegt die *gesetzliche Grundlage* für den Ausgleich zwischen den urheberrechtlichen Ansprüchen des Programmherstellers und den obligatorischen Ansprüchen bzw. den wirtschaftlichen Nutzungsinteressen des Erwerbers in *Art. 12 Abs. 2 URG*<sup>88</sup>. Im Gegensatz zur Weiterentwicklung besteht für die Fehlerbeseitigung kein wirtschaftliches Partizipationsinteresse des Urheberrechtsinhabers: Er soll aus der Fehlerhaftigkeit seines Produktes keine Vorteile ziehen können<sup>89</sup>. Indessen sind die Grenzen zwischen der Behebung von eigentlichen Programmfehlern und Leistungssteigerungen teilweise fließend. Soweit die Fehlerbehebung Einblick in den Sourcecode erfordert, ist immerhin das *Geheimhaltungsinteresse* des Urheberrechtsinhabers zu berücksichtigen<sup>90</sup>.

Das Fehlerbehebungsrecht dürfte auch im schweizerischen Recht analog zur CRRL insoweit *zwingender Natur* sein, als zwar dessen Modalitäten vertraglich geregelt werden können, ein genereller Ausschluss aber unzulässig ist, soweit der Inhaber des Urheberrechts nicht selbst die Fehlerbehebung zu angemessenen Konditionen anbietet, da

OR ein vorgängig oder gleichzeitig erstrittenes Leistungsurteil gegen den Schuldner notwendig ist. Vgl. dazu Basler Kommentar *Wiegand*, N. 6 zu Art. 98 OR, und *Walter Fellmann*, Die Ersatzvornahme nach Art. 98 Abs. 1 OR – «Vollstreckungstheorie» oder «Erfüllungstheorie», recht 1993, S. 109 ff., S. 115 f.

Das *Werkvertragsrecht* enthält in Art. 366 Abs. 2 OR eine spezifische Bestimmung zur Ersatzvornahme bei noch nicht vollendeten Werken. Es ist jedoch umstritten, ob diese analog auch auf die Ersatzvornahme der *Nachbesserung* eines bereits abgelieferten Werkes anwendbar ist. Dies wird vom Bundesgericht in 107 II 50 E. 3 grundsätzlich bejaht. Vgl. dazu differenzierend Basler Kommentar *Zindel/Pulver*, N. 28 zu Art. 366 OR, und *Peter Gauch*, Der Werkvertrag, 4. A., Zürich 1996, N. 1821 ff.

<sup>88</sup> Art. 13a Abs. 1 lit. b VE URG sieht in Analogie zu Art. 5 Abs. 1 CRRL ein *Bearbeitungsrecht* vor, soweit es zur bestimmungsgemäßen Benutzung einschließlich der Fehlerbereinigung notwendig ist.

<sup>89</sup> Vgl. dazu auch *Marly* (Fn. 16), S. 223.

<sup>90</sup> Stellt der Softwarefehler einen *Mangel* im Sinn des Gewährleistungsrechts dar, dürfte der zu erwartende Gebrauch dessen Behebung unabhängig vom Bestand allfälliger Gewährleistungsansprüche im konkreten Fall erlauben. Weil der Inhaber des Urheberrechts nicht zugleich der Vertragspartner des Nutzers sein muss und daher für dessen allfällige Zusicherungen nicht einzustehen hat, ist eine objektiviertete Betrachtungsweise des Mangels vorzunehmen, welcher zur Fehlerbehebung berechtigt. Vgl. dazu auch *Caduff* (Fn. 53), S. 103 f.

andernfalls die Gefahr besteht, dass das Computerprogramm für den Erwerber nutzlos wird<sup>91</sup>.

### 6. Ansprüche auf Herausgabe des Sourcecodes

Aus einem Übertragungs- oder Lizenzvertrag kann hervorgehen, dass der Nutzer in jedem Fall Anspruch auf Herausgabe des Sourcecodes – und eventuell auch der Entwicklungsdokumentation – hat. Wird nur der Objectcode geliefert, liegt in diesem Fall eine bloße *Teilleistung* vor. Der Erwerber kann dementsprechend nach Art. 107 ff. OR vorgehen<sup>92</sup> und auf Lieferung des Sourcecodes klagen<sup>93</sup>.

Wurde die Mitlieferung des Sourcecodes nicht ausdrücklich vereinbart, ist ein entsprechender Anspruch grundsätzlich nur mit Zurückhaltung anzunehmen, da der Erwerber eines Computerprogramms aufgrund der Interessenlage des Urheberrechtsinhabers<sup>94</sup> eine Herausgabe selbst bei Individualsoftware nicht ohne weiteres erwarten darf<sup>95</sup>.

Die Änderung eines Computerprogramms setzt in der Regel Einblick in den Sourcecode voraus<sup>96</sup>. Zumindest bei einem *echten Lizenz-*

<sup>91</sup> Vgl. dazu in Bezug auf die CRRL auch *Czarnota/Hart* (Fn. 16), S. 65 f.

<sup>92</sup> Ein Verzicht auf die ganze Leistung unter *Rückgabe des bereits erhaltenen Objectcodes* ist indessen nur möglich, wenn der bloß teilweise erfüllte Vertrag für den Erwerber objektiv ohne Interesse ist. Vgl. dazu Basler Kommentar *Wiegand*, N. 21 zu Art. 107 OR, mit weiteren Hinweisen. Ein vertragsentscheidendes Interesse am Sourcecode dürfte etwa bei einer Entwicklungslizenz vorliegen.

<sup>93</sup> Die *Klage auf Herausgabe* des Sourcecodes ist *obligatorischer Natur*, was im Konkursfall des Inhabers von Nachteil sein kann. Vgl. zur Absicherung des Konkursrisikos durch eine Escrowvereinbarung weiter hinten III.

<sup>94</sup> Vgl. zu den *Interessen des Urhebers* weiter vorne I.3.

<sup>95</sup> Eine Verpflichtung zur Herausgabe des Sourcecodes wird in der deutschen Literatur überwiegend als nicht interessengerecht verneint, da der Inhaber des Urheberrechts ein erhebliches Geheimhaltungsinteresse hat und eine Mitlieferung nicht üblich ist. Vgl. dazu *Jochen Schneider*, Handbuch des EDV-Rechts, 2. A., Köln 1997, S. 527 und 1774, und *Thomas Hoeren*, Softwareüberlassung als Sachkauf, München 1989, S. 90 ff. Ein *Anspruch auf Mitlieferung* des Sourcecodes dürfte aber zumindest bestehen, wenn sämtliche Rechte an der Software übertragen wurden.

<sup>96</sup> In beschränktem Umfang ist eine unmittelbare Änderung des Objectcodes durch *Patching* möglich. Diese Technik ist allerdings mit erheblichen praktischen Schwierigkeiten und Risiken verbunden, sodass die bloße Ausübung eines Änderungsrechts

*vertrag*<sup>97</sup> mit positiver Rechtsnatur<sup>98</sup> ist der Lizenzgeber verpflichtet, den Lizenznehmer bei der Ausübung der Lizenz zu unterstützen: Umfasst sie auch ein Änderungsrecht, so kann daraus grundsätzlich ein Anspruch auf Zurverfügungstellen des Sourcecodes abgeleitet werden<sup>99</sup>. In BGE 125 III 262 scheint dies vom Bundesgericht im Ergebnis

durch Patching unzumutbar sein dürfte. In bestimmten Fällen können fehlerhafte Ergebnisse oder mangelnde Funktionalitäten auch durch *interoperable Programme* behoben werden (vgl. zu den Voraussetzungen der Dekompilierung für die Schaffung solcher Programme weiter hinten II.7). Gewisse Programme ermöglichen schließlich eine *direkte Anpassung via ihre Benutzeroberfläche*.

<sup>97</sup> Echte Lizenzverträge beruhen in der Grundkonzeption auf einem Verzicht des Lizenzgebers, seine Immaterialgüterrechte gegenüber dem Lizenznehmer geltend zu machen. Beim Erwerb von Software ist allerdings häufig untechnisch von «Lizenz» die Rede, obwohl z.B. ein Kauf- oder Werkvertrag vorliegt.

Auf *Standardsoftware* – welche im Hinblick auf eine größere Anzahl von Kunden ohne individuelle Anpassung geschaffen wurde – ist in der Regel Kaufrecht anwendbar. Vgl. dazu BGE 124 III 456 E. 4b/bb. Der Kauf von Werkexemplaren ist vom Erwerb von Urheberrechten zu unterscheiden. Soweit es nur um einen Art. 12 Abs. 2 URG entsprechenden Gebrauch des Computerprogramms geht, ist der Erwerber nicht auf eine zusätzliche vertragliche Lizenz angewiesen. Zum Kaufvertrag über das Programmexemplar kann jedoch eine Lizenz hinzutreten, wenn dem Erwerber weiter gehende Rechte eingeräumt werden. Vgl. dazu auch *Felix H. Thomann*, Softwareschutz durch das Urheberrecht, in: Softwareschutz, herausgegeben von Felix H. Thomann/Georg Rauber, Bern 1998, S. 36.

Auf Verträge über die Erstellung von *Individualsoftware* sind in der Regel werkvertragsrechtliche Bestimmungen anwendbar. Vgl. dazu *Ursula Widmer*, Risikofolgeverteilung bei Informatikprojekten: Haftung für Softwaremängel bei Planung und Realisierung von Informationssystemen, Diss. Bern 1989, S. 46 ff., *Allen Fuchs*, Vertrag über die Entwicklung von Software, in: EDV-Software: Rechtsschutz Vertragswesen-Checklisten, herausgegeben von Ivan Cherpillod, Lausanne 1986, S. 33 ff., S. 39 f., sowie *Gauch* (Fn. 87), Rz. 334. *Doris Slongo Wagen*, Der Softwareherstellungsvertrag, Diss. Zürich 1991, S. 155 f., betrachtet den Softwareherstellungsvertrag hingegen generell als Innominatkontrakt *sui generis*.

<sup>98</sup> Vgl. zur positiven bzw. negativen *Natur von Lizenzverträgen* differenzierend *Hilaty*, Lizenzvertragsrecht (Fn. 69), S. 7 ff., *von Büren*, SIWR II/1, S. 240, und *Conrad Weinmann*, Die Rechtsnatur der Lizenz, Bern 1996, zugleich Diss. Zürich 1996, S. 587 ff., mit weiteren Hinweisen.

<sup>99</sup> Vgl. dazu bejahend auch *Neff/Arn*, SIWR II/2, S. 270. Nach einem Entscheid des deutschen BGH aus dem Jahr 1986 (NJW 1987, S. 1259 f.) ist für einen Anspruch auf Herausgabe des Sourcecodes entscheidend, ob der Erwerber nach dem Vertrag selbst Programmänderungen und Fehlerbeseitigung vornehmen sollte. Vgl. dazu auch *Jochen Marly*, Softwareüberlassungsverträge, Erscheinungsformen, Leistungsstörung, Vertragsgestaltung, allgemeine Geschäftsbedingungen, 2. A., München 1997, Rz. 58 ff., und *Matthias Brandi-Dohrn*, Gewährleistung bei Hard- und Softwaremängeln, 2. A., München 1994, S. 33.

allerdings verneint zu werden. Inwieweit aus diesem Entscheid allgemein gültige Schlüsse gezogen werden können, ist fraglich<sup>100</sup>. Ein Änderungsrecht, welches sich nur auf den Objectcode bezieht, ist jedenfalls oft praktisch nutzlos.

Besteht ein Anspruch auf Herausgabe des Sourcecodes auch zur *Fehlerbehebung*? Solange vertragliche Gewährleistungsansprüche bestehen, fehlt in der Regel ein Herausgabeinteresse. Hat der Erwerber Nachbesserungsansprüche, welche durch den Vertragspartner nicht erfüllt werden, besteht eventuell ein Recht auf *Ersatzvornahme*<sup>101</sup>. Diese ist praktisch oft nur durchführbar, wenn der dazu Berechtigte über den Sourcecode (und eventuell auch die Entwicklungsdokumentation) verfügt. Im Rahmen von Art. 98 OR ist indessen keine Herausgabe von Sachen vorgesehen<sup>102</sup>, sodass sich ein allfälliger Herausgabeanspruch auf das konkrete Vertragsverhältnis abstützen lassen muss<sup>103</sup>.

<sup>100</sup> In dem BGE 125 III 262 zugrundeliegenden Sachverhalt wurde in einem Lizenzvertrag der Lizenznehmerin ein umfassendes Abänderungsrecht eingeräumt, ohne ihr allerdings Eigentum an einem Programmexemplar einzuräumen. In einem Nachtrag zum Vertrag wurde bestimmt, dass nur der Objectcode installiert werde. Schließlich wurde aus nicht näher geklärten Gründen durch die Lizenzgeberin dennoch auch der Sourcecode mitinstalliert. Diese hat sich gegen spätere Veränderungen des Sourcecodes seitens der Lizenznehmerin mit urheberrechtlichen Ansprüchen zur Wehr gesetzt. Im Lizenzvertrag war ein Vorrang von Nachträgen vorgesehen. Das Bundesgericht schließt aus der alleinigen Erwähnung des Objectcodes im Nachtrag, dass praktisch ein nachträglicher Verzicht auf die Ausübung des ursprünglich vereinbarten Änderungsrechts stattgefunden habe. Es fragt sich immerhin, ob die Lizenznehmerin nicht davon ausgehen durfte, dass die Lizenzgeberin durch die Mitlieferung des Sourcecodes den Widerspruch zugunsten des ursprünglichen Vertrages lösen wolle, bzw. ob das Verhalten nach Vertragsschluss Indizien für die Auslegung des ursprünglichen Vertragswillens enthält. Vgl. dazu auch *Chappuis* (Fn. 65), S. 476.

<sup>101</sup> Vgl. dazu Fn. 87.

<sup>102</sup> Vgl. zur Rechtsnatur von Art. 98 Abs. 1 OR *Fellmann* (Fn. 87), S. 109 ff. Aus Art. 98 OR, können jedenfalls keine *Nebenleistungspflichten* auf Herausgabe des Sourcecodes abgeleitet werden. Hingegen können sich solche Nebenleistungspflichten aus dem Vertragsverhältnis ergeben. Vgl. dazu *Fröhlich-Bleuler* (Fn. 47), S. 575.

<sup>103</sup> Beim vorzeitigen Abbruch eines Entwicklungsprojekts mit *Werkvertragscharakter* kann eventuell aus Art. 366 Abs. 2 OR ein Herausgabeanspruch der bereits entwickelten Teile abgeleitet werden. Es fragt sich allerdings, ob der Herausgabeanspruch weiter als der Ablieferungsanspruch bei Vollendung des Werks geht, d.h. ob der unfertige Sourcecode auch herausverlangt werden kann, wenn bei Werkvollendung kein Anspruch bestanden hätte. Dafür spricht, dass Sourcecode und Entwicklungsdokumenta-

## 7. Analyse des Objectcodes

Ist der Sourcecode eines Computerprogramms nicht verfügbar, kann versucht werden, durch *reverse engineering*<sup>104</sup> anhand des Objectcodes auf seinen Inhalt zurückzuschließen. Solche Maßnahmen sind indessen technisch aufwändig und erlauben praktisch nie eine vollständige Rekonstruktion des ursprünglichen Sourcecodes<sup>105</sup>, was für Programmänderungen allerdings auch nicht in jedem Fall notwendig ist.

*Triviale Programme*, welche die Schranke der Individualität noch nicht erreichen<sup>106</sup>, dürfen unbeschränkt analysiert, dekompiert<sup>107</sup> und verwertet werden, weil das Urheberrecht auf sie nicht anwendbar ist<sup>108</sup>. Ebenso dürfen triviale Teile eines nur teilweise individuellen Programms dekompiert und verwertet<sup>109</sup> werden<sup>110</sup>.

Die Analyse urheberrechtlich geschützter Programme ist hingegen nicht ohne weiteres zulässig. Da jeder Ladevorgang eines Computerprogramms technisch gesehen die Herstellung einer vorübergehenden Programmkopie im Arbeitsspeicher des Computers mit sich bringt,

tion zwingend nötig sind, wenn das Werk durch einen Dritten vollendet werden soll, selbst wenn der Besteller im Fall der ordnungsgemäßen Fertigstellung darauf nicht angewiesen gewesen wäre.

<sup>104</sup> Vgl. zu den verschiedenen Arten des *reverse engineering* Marly (Fn. 16), S. 269 ff. Neben der nachfolgend unter II.8 behandelten Dekompilierung kommt insbesondere *line tracing* in Betracht: mit einem Hilfsprogramm wird das Computerprogramm schrittweise ausgeführt und die Wirkung jedes Programmschritts auf den Speicherinhalt untersucht.

<sup>105</sup> Die Umwandlung des Objectcodes erfolgt in aller Regel nicht in die ursprüngliche Programmiersprache, sondern in *Assembler*. Informationen, welche bereits bei der Kompilierung verloren gegangen sind, aber für die Weiterentwicklung des Programms unter Umständen von entscheidender Bedeutung sind (z.B. Bemerkungen des Programmierers zum Programmaufbau), können nicht wiederhergestellt werden.

<sup>106</sup> Vgl. zu den urheberrechtlichen Schutzvoraussetzungen für Computerprogramme weiter vorne II.1.

<sup>107</sup> Vgl. dazu nachfolgend unter II.8.

<sup>108</sup> Vgl. dazu auch *Czarnota/Hart* (Fn. 16), S. 43. Falls nicht ausnahmsweise andere Indizien für das Fehlen urheberrechtlichen Schutzes vorliegen, ermöglicht allerdings meistens erst das Ergebnis einer Dekompilierung *ex post* die – oft nicht einmal eindeutige – Beurteilung der Anwendbarkeit des Urheberrechts. Vgl. dazu auch *Xavier Linant de Bellefonds*, *Le droit de décompilation des logiciels: une aubaine pour les cloneurs ?* *La Semaine Juridique I/1998*, S. 479 ff., S. 480 f., S. 484.

<sup>109</sup> Vgl. dazu immerhin Art. 5 lit. c UWG.

<sup>110</sup> Vgl. dazu auch *Huet* (Fn. 19), S. 316 f.

tangiert bereits das bloße *Ablaufenlassen zur Beobachtung* das Urheberrecht<sup>111</sup>. Die Rekonstruktion des Sourcecodes erfordert zusätzlich eine Veränderung des Objectcodes<sup>112</sup>. Es fragt sich daher, woraus sich solche, das Urheberrecht des Herstellers einschränkende Rechte (Lizenzen) ableiten lassen. Das schweizerische Urheberrecht enthält ähnlich wie die CRRL beschränkte *gesetzliche Lizenzen* einerseits für die *Programmbesichtigung* und -analyse ohne Codeänderung<sup>113</sup>, andererseits für die Wiederherstellung des Sourcecodes durch *Dekompilierung*<sup>114</sup>.

Der zulässige Gebrauch eines Computerprogramms umfasst das Untersuchen und Testen zur Ermittlung der einem Programmelement zugrundeliegenden Ideen und Grundsätze<sup>115</sup>. Dieses Recht beruht auf dem Grundsatz, dass nur die Ausdrucksform, nicht aber die einem Werk zugrundeliegenden Ideen<sup>116</sup> urheberrechtlich schützbar sind. Bereits aus der komplizierten Formulierung von Art. 17 Abs. 1 URV wird allerdings deutlich, dass die Programmanalyse nur in sehr eingeschränktem Maß erlaubt werden soll<sup>117</sup>. Einerseits erstreckt sie sich nicht auf die *Struktur* des ganzen Programms sondern ist auf einzelne

<sup>111</sup> Vgl. dazu weiter vorne II.3.

<sup>112</sup> Anstelle der für Menschen extrem unübersichtlichen binären Darstellung zeigen selbst einfachste Beobachtungswerkzeuge Programmbefehle in hexadezimaler Form bzw. in Assemblersprache an. Die beobachteten Programmteile werden also auch in diesem Fall in eine andere Form übersetzt, was grundsätzlich die Urheberrechte des Herstellers tangiert. Vgl. dazu *Marly* (Fn. 16), S. 269.

<sup>113</sup> Art. 17 Abs. 1 URV/Art. 5 Abs. 3 CRRL.

<sup>114</sup> Art. 21 URG und 17 Abs. 2 URV/Art. 6 lit. c CRRL.

<sup>115</sup> Nach *Fröhlich-Bleuler* (Fn. 47), S. 577 hat es der schweizerische Gesetzgeber versäumt, die *zwingende Natur des Beobachtungsrechts* festzuhalten. Ein Wegbedingen dieser Minimalrechte dürfte allerdings in Konflikt mit dem essenziellen bestimmungsgemäßen Gebrauch geraten. Vgl. auch § 42d des österreichischen UrhG, wonach das Beobachtungsrecht zwingenden Charakter hat.

<sup>116</sup> Vgl. zur Abgrenzung von Ausdrucksformen und zugrundeliegenden *Ideen* Fn. 16.

<sup>117</sup> Die Bestimmungen über *reverse engineering* und Dekompilierung bezwecken einen umfassenden Geheimnisschutz von Computerprogrammen, der dem übrigen Urheberrecht fremd ist. Das Urheberrecht scheint sich hier im Ergebnis zu einem *Know-how-Schutz von Immaterialgütern* zu entwickeln. Vgl. in diesem Zusammenhang auch Art. 70a VE URG, welcher den strafrechtlichen Schutz technischer Maßnahmen bezweckt.

Programmelemente beschränkt<sup>118</sup>. Andererseits muss sie im Rahmen der *Handlungen zur bestimmungsgemäßen Verwendung* des Computerprogramms erfolgen<sup>119</sup>. Mit den in Art. 17 Abs. 1 lit. a URV bzw. Art. 5 Abs. 3 CRRL erwähnten Handlungen des Ladens, Anzeigens, Ablaufens, Übertragens und Speicherns ist eine Analyse der einem Computerprogramm zugrundeliegenden Ideen nur in sehr geringem Maß möglich. Das Beobachtungsrecht ist somit zu einer effizienten Änderung des Computerprogramms in der Regel nicht ausreichend.

### 8. Rekonstruktion des Sourcecodes

Die *Dekompilierung* stellt technisch gesehen wohl die effizienteste Möglichkeit dar, Rückschlüsse auf den ursprünglichen Sourcecode zu ziehen. Durch die Dekompilierung entsteht eine Kopie des Objectcodes in einer anderen Form (*Assemblersprache*). Da jeder Kopier- und Änderungsvorgang die Urheberrechte tangiert, ist dazu grundsätzlich die Zustimmung des Inhabers des Urheberrechts oder eine gesetzliche Erlaubnis notwendig.

Der heutige Art. 21 URG ist unter dem Eindruck von Art. 6 CRRL<sup>120</sup> im Nationalrat in das geltende URG eingefügt worden<sup>121</sup>. Im

<sup>118</sup> Nach *Marly* (Fn. 16), S. 272, und *Czarnota/Hart* (Fn. 16), S. 70, muss zwischen der zulässigen Beobachtung der *Programmfunktion* und der nicht zulässigen Untersuchung des *Programms* selbst unterschieden werden.

<sup>119</sup> Art. 13a Abs. 2 VE URG ist kongruent mit Art. 5 Abs. 3 CRRL. Demgegenüber wird im geltenden Art. 17 Abs. 1 lit. b URV nicht gesagt, was unter den *Handlungen zur bestimmungsgemäßen Verwendung* zu verstehen ist. Diese werden in lit. a der selben Bestimmung analog präzisiert, ohne allerdings explizit festzuhalten, ob die Aufzählung abschließenden Charakter hat. *Neff/Arn*, SIWR II/2, S. 256, und *Czarnota/Hart* (Fn. 16), S. 69, gehen davon aus, dass Techniken wie die Dekompilierung ausgeschlossen sind, welche Programmveränderungen nötig machen.

<sup>120</sup> Vgl. zu den *Hintergründen* von Art. 6 CRRL auch *Josette Beer-Gabel/Régis Chémain*, La décompilation des logiciels: l'industrie européenne face au droit d'auteur, *Revue Trimestrielle de Droit Européen*, S. 363 ff., sowie *Robert J. Hart*, Interfaces, Interoperability and Maintenance, *European Intellectual Property Review* 1991, S. 111 ff. *Linant de Bellefonds* (Fn. 108), S. 483, ist der Ansicht, dass die Dekompilierung ein Recht *sui generis* mit wettbewerbsrechtlichem Charakter darstelle.

<sup>121</sup> Vgl. dazu *Carlo Govoni*, Der urheberrechtliche Schutz von Computerprogrammen, *AJP* 1993, S. 569 ff., S. 570, sowie *Thomann* (Fn. 20), S. 563 ff.

schweizerischen Recht ist – wie auch in der CRRL – nur die *Dekompilierung*<sup>122</sup> von *Interfaces*<sup>123</sup> zur *Beschaffung von Schnittstelleninformationen*<sup>124</sup> explizit geregelt<sup>125</sup>.

Die Dekompilierung setzt zunächst das *Recht* voraus, das *Computerprogramm zu gebrauchen*<sup>126</sup>. Durch Art. 21 URG werden somit primär die Interessen der Programmbenutzer, nicht aber diejenigen der anderen Marktteilnehmer geschützt. Der Berechtigte braucht die Dekompilierung nicht selbst vorzunehmen, sondern kann Dritte an seiner Stelle damit beauftragen<sup>127</sup>.

<sup>122</sup> Das geltende URG spricht anders als die CRRL nicht von «Dekompilierung», sondern von «*Entschlüsselung*» (*décryptage*). Allein durch Dekompilierung kann nicht unbedingt ein brauchbarer Sourcecode gewonnen werden. In der Regel sind weitere Arbeitsschritte zur Rekonstruktion des Sourcecodes nötig. Einige Hersteller setzen zudem kryptografische Techniken ein, um die Dekompilierung zu erschweren. Im Rahmen von Art. 21 URG dürfen über die eigentliche Dekompilierung hinaus weitere Verfahren angewendet werden, welche zur Gewinnung der Schnittstelleninformationen notwendig sind.

<sup>123</sup> Unter *Interface* werden hier diejenigen Teile eines Computerprogramms verstanden, welche notwendig sind, um Daten mit anderen Programmen bzw. Unterprogrammen (Objekten) austauschen. M.E. ist für die Dekompilierung nicht von vorneherein notwendig, dass diese Schnittstellen vom Hersteller im Hinblick auf den Datenaustausch mit interoperablen Programmen konzipiert wurden (offene Schnittstellen). Wohl a.M. hingegen *Linant de Bellefonds* (Fn. 108), S. 480 f.

<sup>124</sup> Der Begriff der *Schnittstelleninformation* ist gegenüber demjenigen des *Interfaces* weiter gefasst, indem er auch im Programm selbst nicht enthaltene Informationen umfasst, welche für das Funktionieren des Datenaustausches notwendig sind. Schnittstelleninformationen haben oft einen erheblichen kommerziellen und strategischen Wert: Nur wer über sie verfügt, kann interoperable Programme herstellen und damit eventuell auch entsprechende Produkte des Ausgangsprogramms konkurrenzieren. Die Kontrolle der Schnittstelleninformation hat somit *wettbewerbsrechtliche Relevanz*. Die EG-Kommission hat sich bereits 1984 mit einem Fall beschäftigt, in welchem IBM Hardwareschnittstellen für ein modular aufgebautes Rechnersystem erst nach der Produktauslieferung bekannt gab und dadurch die zeitgerechte Entwicklung von interoperablen Produkten behinderte. Vgl. dazu EG Bull. Nr. 7/8 1984, S. 7 ff. und Nr. 10 1984, S. 105 ff.

<sup>125</sup> Art. 21 URG in Verbindung mit Art. 17 Abs. 2 URV. Art. 21 Abs. 4 VE URG stellt klar, dass auf das Dekompilierungsrecht nicht verzichtet werden kann.

<sup>126</sup> Art. 12 Abs. 2 URG. Vgl. dazu Fn. 49 und 59.

<sup>127</sup> Art. 21 Abs. 1 URG. Hingegen darf ein vom Erwerber mit der Dekompilierung beauftragter *Dritter* die gewonnenen Informationen nicht selbst weiterverwerten (Art. 21 Abs. 2 URG und 6 Abs. 2 CRRL).

Art. 21 Abs. 1 URG erlaubt nur die Dekompilierung der *erforderlichen*<sup>128</sup> Informationen. Die Auffindung der relevanten Schnittstelleninformation innerhalb eines größeren Objectcodes ist allerdings mit technischen Schwierigkeiten verbunden. Selbst wenn es gelingt, eine bestimmte Schnittstelle zu lokalisieren, enthält das Interface selbst nicht unbedingt alle für die Interoperabilität notwendigen Informationen, da z.B. erläuternde Bemerkungen des Programmierers über Zusammenhänge mit anderen Programmteilen im Objectcode fehlen. Unter Umständen müssen daher über das eigentliche Interface hinaus weitere Programmteile analysiert werden<sup>129</sup>.

Die zu dekompilierenden Schnittstelleninformationen müssen für den Entwickler des interoperablen Programms *nicht bereits* «ohne weiteres» zugänglich sein<sup>130</sup>. Recherchen nach dem Inhaber des Urheberrechts oder allfälligen anderen Publikationsquellen dürften daher nicht zumutbar sein<sup>131</sup>. Der Urheberrechtinhaber ist zu einer Bekanntgabe unter Vorbehalt wettbewerbsrechtlicher Bestimmungen

<sup>128</sup> Art. 6 Abs. 1 der CRRL erlaubt die Dekompilierung nur, soweit sie zur Herstellung der Interoperabilität *unerlässlich* (indispensable) ist. Vgl. dazu *Czarnota/Hart* (Fn. 16), S. 77. Art. 21 Abs. 1 lit. a VE URG stimmt hier mit der CRRL überein. Die deutsche und italienische Fassung des geltenden Art. 21 URG sprechen von den «erforderlichen» Informationen (informazioni necessarie per l'interfaccia), währenddem in der französische Fassung ohne weitere Einschränkung von «informations sur des interfaces» die Rede ist. Art. 17 URV versucht allerdings das URG konform zur CRRL zu präzisieren indem sie das Erfordernis der Unerlässlichkeit zur Herstellung der Interoperabilität hingegen in allen Sprachen erwähnt: «... les informations nécessaires sur les interfaces sont celles qui sont indispensables à l'élaboration de l'interopérabilité.»

<sup>129</sup> Damit der tatsächliche Zugang zur Schnittstelleninformation nicht von vornherein illusorisch wird, muss das Programm soweit analysiert werden können, wie es die Auffindung der Schnittstelleninformation erfordert. Vgl. dazu auch *Czarnota/Hart* (Fn. 16), S. 77, ablehnend *Linant de Bellefonds* (Fn. 108), S. 482. Sollte das Interesse des Nutzers an der Schnittstelleninformation aufgrund des Umfangs der notwendigen Dekompilierung in einem *Missverhältnis zu den Geheimhaltungsinteressen* des Urhebers stehen, enthalten Art. 21 Abs. 2 URG und Art. 6 Abs. 3 CRRL immer noch ein Ventil.

<sup>130</sup> Art. 17 Abs. 2 URV/Art. 1 Abs. 6 lit. b CRRL.

<sup>131</sup> Nach *Marly* (Fn. 16), S. 319, Fn. 315 (mit Hinweisen auf die Entstehungsgeschichte) muss im EU-Raum nicht vorgängig der *Inhaber des Urheberrechts kontaktiert* werden. Vgl. dazu auch *Alfred P. Meijboom* in: Copyright Software Protection in the EC, herausgegeben von Herald D. J. Jongen/Alfred P. Meijboom, Deventer 1993, S. 16. Praktische Überlegungen sprechen trotzdem für diesen Weg: Einerseits ist der technische Aufwand der Dekompilierung in der Regel erheblich. Andererseits dient eine entsprechende Anfrage zur Beweissicherung der Unzugänglichkeit.

nicht verpflichtet, muss aber mit der Dekompilierung rechnen, wenn er sie nur unvollständig oder zu einschränkenden Bedingungen liefert.

Die *Verwendung* der durch Dekompilierung gewonnenen Schnittstelleninformation ist nur zu Entwicklung, Wartung und Gebrauch von interoperablen Computerprogrammen<sup>132</sup> zulässig. Soweit die dekompliierten Informationen urheberrechtsfähig sind, setzt ihre Verwendung unter Vorbehalt der freien Benutzung<sup>133</sup> die Zustimmung des Urheberrechtsinhabers voraus<sup>134</sup>.

Schließlich ist die Verwendung der Schnittstelleninformation an die Voraussetzung gebunden, dass dadurch weder die *normale Auswertung des Programms* noch die *rechtmäßigen Interessen der Rechtsinhaber*

<sup>132</sup> Namentlich bei Betriebssystemen kann ein Interesse an der Dekompilierung von Schnittstellen für *interoperable Hardware* (z.B. Ausgabegeräte) bestehen. Die Grenzen zwischen Hard- und Software sind fließend. Insbesondere können Computerprogramme in Hardwarechips gespeichert oder an Hardwarebausteine gebunden werden. Im Hinblick auf den vom schweizerischen Gesetzgeber und vom europäischen Richtliniengeber verfolgten Zweck der freien Interoperabilität, welcher das baukastenartige Zusammenstellen von IT-Systemen aus Komponenten verschiedener Hersteller erlauben soll, rechtfertigt sich eine weite Auslegung, welche auch die Schaffung interoperabler Hardware erlaubt. Vgl. dazu etwa *Marly* (Fn. 16), S. 322 ff., und *Lehmann* (Fn. 19), S. 334, a.M. hingegen *Linant de Bellefonds* (Fn. 108), S. 481, *Czarnota/Hart* (Fn. 16), S. 82, und *Haberstumpf* (Fn. 24), S. 162, Rz. 171.

<sup>133</sup> Die urheberrechtlichen Abwehrensprüche gegenüber Werkverwendungen werden durch das Prinzip der *freien Benutzung* relativiert, wonach schutzfähige Elemente eines bestehenden Werkes ausnahmsweise frei übernommen werden können, wenn ihre Bedeutung angesichts der Individualität des neuen Werks als völlig untergeordnet erscheint. Vgl. dazu *Cherpillod* (Fn. 16), S. 145 ff., und ders., *SIWR II/1*, S. 277 ff., *Barrelet/Egloff* (Fn. 17), N. 12 zu Art. 11 URG, und *Troller* (Fn. 16), Band I, S. 259, ablehnend hingegen *Dessemontet* (Fn. 16), Rz. 400 ff. Der Grundsatz der freien Benutzung wurde in BGE 125 III 328 E. 4c unter dem neuen URG bestätigt. In solchen Fällen ist im Ergebnis eine Abwägung zwischen den Interessen beider Schöpfer vorzunehmen.

<sup>134</sup> Vgl. dazu im Hinblick auf den 10. Erwägungsgrund der CRRL hingegen *Haberstumpf* (Fn. 24), S. 163, Rz. 173. Beruht die Ausgestaltung eines Interfaces auf technischen Notwendigkeiten, ist nach der hier vertretenen Auffassung *Individualität* ausgeschlossen, sodass es ohne weiteres übernommen werden darf. Zum gleichen Ergebnis kommen auch *Marly* (Fn. 16), S. 324, und *Czarnota/Hart* (Fn. 16), S. 82. Vom eigentlichen Interface zu unterscheiden sind dessen *Spezifikationen* (z.B. das Format der auszutauschenden Daten). Diese dürfen auf jeden Fall übernommen werden, da andernfalls keine Interoperabilität möglich ist.

ber unzumutbar beeinträchtigt werden<sup>135</sup>. Es soll primär die Einsparung von Eigenentwicklungskosten durch Dekompilierung vermieden werden, welche zu Wettbewerbsverzerrungen führen könnte<sup>136</sup>. Hingegen ist die Schnittstellendekompilierung auch zur Erleichterung des Datenaustausches mit konkurrierenden Programmen grundsätzlich zulässig<sup>137</sup>.

Ein legitimes Interesse an einer Programmanalyse durch Dekompilierung kann auch *aus anderen Gründen* als der Schaffung von interoperablen Programmen bestehen, z.B. zur Fehlerbehebung<sup>138</sup>, zum Auffinden von Sicherheitslücken, der Untersuchung der Wirkungsweise von Viren oder bei Verdacht der Übernahme eines eigenen Programms<sup>139</sup>. Es fragt sich daher, ob Art. 21 URG eine abschließende Regelung der zulässigen Dekompilierung enthält oder ob weitere Fälle aus allgemeinen Grundsätzen<sup>140</sup>, insbesondere aus dem gesetzlichen

<sup>135</sup> Art. 21 Abs. 2 URG/Art. 6 Abs. 3 CRRL. Nicht alle wünschbaren Schnittstellen zu bestehenden Programmen sind technisch notwendig: Gewisse Funktionen können anstatt durch Verknüpfung mit einem bestehenden Programm auch anderweitig sichergestellt werden. Der Entwickler des Zweitprogramms braucht allerdings einen gewissen Ermessensspielraum, welche Funktionen er nutzen will. Sein Programm muss indessen *unabhängig entwickelt* sein und sich nicht auf die bloße Anknüpfung an das Originalprogramm beschränken.

<sup>136</sup> Art. 17 Abs. 2 URV und Art. 21 Abs. 2 lit. c VE URG nehmen auf Art. 6 Abs. 2 lit. c der CRRL Bezug, wonach die Informationen nicht für die Entwicklung, Herstellung oder Vermarktung eines Programms mit im Wesentlichen *ähnlicher Ausdrucksform* verwendet werden dürfen. Offenbar sind dort Programme mit ähnlichem Aufbau gemeint. Vgl. dazu *Marly* (Fn. 16), S. 316 f. Diese sind indessen oft schon urheberrechtsverletzend oder nicht «unabhängig entwickelt».

<sup>137</sup> Vgl. dazu *Marly* (Fn. 16), S. 316 f., *Czarnota/Hart* (Fn. 16), S. 82, und *Lehmann* (Fn. 19), S. 334, ders. (Fn. 47), S. 17, sowie ablehnend *Linant de Bellefonds* (Fn. 108), S. 481. Für ein neues Textverarbeitungsprogramm mit neuartigen Funktionalitäten ist es z.B. von großer praktischer Bedeutung, dass Dokumente, welche mit anderen Programmen geschrieben wurden, damit weiter bearbeitet werden können.

<sup>138</sup> *Marly* (Fn. 16), S. 314 ff. und 329.

<sup>139</sup> Vgl. dazu *Marly* (Fn. 16), S. 277, *Manfred Rehbinder*, Urheberrecht, 11. A., München 2001, S. 200, und *Kai Vinck* in: Friedrich Karl Fromm/Wilhelm Nordemann/Kai Vinck/Paul W. Hertin, Urheberrecht, Kommentar zum Urheberrechtsgesetz und zum Urheberrechtswahrnehmungsgesetz, 9. A., Stuttgart/Berlin/Köln 1998, N. 4 zu § 69e des deutschen UrhG.

<sup>140</sup> Vgl. auch zur Dekompilierung im Rahmen des US-amerikanischen *fair-use-Prinzips* *Sega Enterprises Ltd. vs. Accolade Inc.*, 977 F2d 1510, 9th Cir. (1992), sowie *Marly* (Fn. 16), S. 295 mit weiteren Hinweisen und *Schneider* (Fn. 95), S. 426 ff., Rz. 256 ff.

Gebrauchsrecht abgeleitet werden können. Die Materialien ergeben keine eindeutigen Anhaltspunkte<sup>141</sup>. Die Anwendungsbereiche der beiden Bestimmungen überschneiden sich nur teilweise, sodass Art. 21 URG wohl nur partiell den Charakter einer *lex specialis* gegenüber Art. 12 Abs. 2 URG hat<sup>142</sup>. Ist etwa im Rahmen einer Fehlerbehebung, einer zulässigen Änderung oder zu Sicherheitszwecken die Untersuchung eines Programmelements notwendig, so sollte zur Dekompilierung geschritten werden dürfen, wenn die erforderlichen Informationen anders nicht rechtzeitig beschafft werden können<sup>143</sup>. Hingegen dürfte die umfassende Analyse eines Programmes im Hinblick auf Piraterieverdacht oder aus bloßer (wissenschaftlicher) Neugier kaum unter dessen bestimmungsgemäßen Gebrauch subsumiert werden können<sup>144</sup>.

Es fragt sich, ob die Dekompilierung unabhängig von der urheberrechtlichen Qualifikation durch das *Lauterkeitsrecht* beschränkt wird<sup>145</sup>. Das UWG ist im Gegensatz zum URG auf wettbewerbsrelevante Handlungen ausgerichtet und verpönt daher grundsätzlich nicht die Produkteanalyse selbst, sondern die Auswertung von verbotenerweise erfahrenen Geheimnissen<sup>146</sup>. Art. 6 UWG schützt «ausgekundschafte oder sonst wie unrechtmäßig erfahrene Geschäfts- und Fabrikationsgeheimnisse» vor der Verwertung und Weitergabe. Soweit die

<sup>141</sup> Vgl. Amtl. Bull. NR 1992, S. 4 ff. Vgl. für die CRRL hingegen wohl ablehnend den gemeinsamen Standpunkt des Rates vom 13. Dezember 1990 im Hinblick auf die Genehmigung einer Richtlinie über den Rechtsschutz von Computerprogrammen, GRUR Int. 1991, S. 548.

<sup>142</sup> Nach *Günther* (Fn. 66), S. 327, ist auch die auf der CRRL basierende Dekompilierungsregelung von § 69e des deutschen UrhG *nicht abschließend*, sodass für die nicht erfassten Fälle analog zur Änderungsbefugnis eine Interessenabwägung vorzunehmen ist. Vgl. zur Rechtslage in der EU auch *Marly* (Fn. 16), S. 314 ff., *Huet* (Fn. 19), S. 316 f., *Beer-Gabel/Chemain* (Fn. 19), S. 376, *Czarnota/Hart* (Fn. 16), S. 65 und 71, und *Schneider* (Fn. 95), S. 499 ff., Rz. 472 ff.

<sup>143</sup> Wenn der Inhaber des Urheberrechts einer vertraglichen Verpflichtung zur Herausgabe des Sourcecodes nicht nachkommt, lässt sich die Dekompilierung wohl nicht direkt auf Art. 98 OR abstützen (vgl. dazu weiter vorne II.6). Falls eine ersatzweise Dekompilierung die Voraussetzungen von Art. 52 Abs. 3 OR erfüllt, entfallen indessen allfällige *Schadenersatzansprüche* des Urhebers.

<sup>144</sup> Vgl. dazu zum deutschen Recht auch *Haberstumpf* (Fn. 24), S. 161.

<sup>145</sup> Vgl. zum Verhältnis zwischen Urheber- und Lauterkeitsrecht auch Fn. 28.

<sup>146</sup> Vgl. dazu auch Fn. 28, sowie zum deutschen Recht *Marly* (Fn. 16), S. 302 f.

Dekompilierung urheberrechtlich zulässig ist, stellt sie kein Auskundschaften dar. Nur wenn es bereits es an der urheberrechtlichen Zulässigkeit fehlt, kann das *reverse engineering* allenfalls unter das Tatbestandselement «sonstwie unrechtmäßig erfahren» subsumiert werden. Das Lauterkeitsrecht schränkt somit das Recht zur Dekompilierung gegenüber dem Urheberrecht nicht weiter ein.

### III. Escrow und Hinterlegung

Um sowohl den Interessen des Urheberrechtsinhabers an der Geheimhaltung des Sourcecodes als auch denjenigen des Erwerbers in Situationen Rechnung zu tragen, in welchen er besonders auf dessen Herausgabe angewiesen ist, kann vertraglich die Hinterlegung (*Escrow*) bei einem Dritten vereinbart werden. Die Escrowverpflichtung wird oft in einer Rahmenbestimmung in einem anderen Vertrag (z.B. Entwicklungs- und Lizenzvertrag) vorgesehen und durch einen späteren *Escrowagreement* näher präzisiert. Eine umfassende Darstellung des Escrowrechts würde den Rahmen dieses Aufsatzes bei weitem sprengen, sodass hier nur auf einzelne zentrale Aspekte hingewiesen werden kann.

Der Hinterleger verpflichtet sich im Escrowvertrag, Kopien des Sourcecodes bei einem Dritten (Escrowagent) einzuliefern und regelmäßig zu aktualisieren. Der Escrowagent verspricht, den Sourcecode (und eventuell weitere Dokumente) sicher zu verwahren und nur in den vertraglich vorgesehenen Fällen an den Lizenznehmer herauszugeben<sup>147</sup>. Oft erhält er die Stellung eines fiduziarischen Eigentümers<sup>148</sup>.

<sup>147</sup> Diese Grundkonzeption kann zu einem *Mehrparteienverhältnis* erweitert werden, indem verschiedene Urheberrechtsinhaber und beliebig viele Lizenznehmer mit einbezogen werden. Vgl. zu den verschiedenen Varianten *Kurt U. Blickenstorfer*, *Der Sourcecode-Escrow*, in: *Softwareschutz*, hrsg. von Felix H. Thomann/Georg Rauber, Bern 1998, S. 212 ff., S. 220 ff. Ansprüche auf Herausgabe des Sourcecodes können auch zugunsten von Dritten vorgesehen werden, welche nicht Vertragspartei sind.

<sup>148</sup> Es ist sowohl möglich, dass dem Escrowagenten *bloßer Besitz* eingeräumt wird und das Eigentum beim Einlieferer bleibt, als auch dass er *fiduziarisches Eigentum* erhält. Vgl. dazu *Stefan Gerster*, *Das Escrow Agreement als obligationenrechtlicher Vertrag*, Diss. Zürich 1991, S. 72 ff., und *Blickenstorfer* (Fn. 147), S. 234 ff.

Die Qualität von Escrowlösungen hängt vor allem von zwei kritischen Punkten ab: Wie sichergestellt wird, dass der Lizenzgeber tatsächlich jeweils die *relevanten Dokumente* liefert<sup>149</sup> und unter welchen *Voraussetzungen* sie an den Lizenznehmer herausgegeben werden dürfen<sup>150</sup>.

Eine Hinterlegung kommt auch im *Interesse des Urheberrechtinhabers* in Betracht, wenn er den Sourcecode dem Erwerber mitliefert: Durch Hinterlegung des Sourcecodes bei einem Notar oder einem Escrowagenten kann im Hinblick auf künftige Prozesse beweismässig gesichert werden, wann ein Programm bzw. eine bestimmte Version erarbeitet wurde.

#### IV. Ergebnisse

Computerprogramme werden meist nur in der Form des für Menschen unverständlichen *Object- bzw. Binärcodes* geliefert. Zur Analyse und Änderung eines Programms sowie zur Schaffung interoperabler Software sind jedoch in der Regel im *Sourcecode* enthaltene Informationen notwendig. Bei Vorliegen einer vertraglichen Änderungslizenz kann ein – vom Bundesgericht allerdings noch nicht anerkannter – Anspruch auf Herausgabe des nicht mitgelieferten Sourcecodes bestehen. Wegen seiner zentralen Bedeutung für die Weiterentwicklung des Programms haben unter Umständen sowohl Urheberrechtinhaber als auch Nutzer legitime Interessen am Sourcecode. Durch *Escrowverträge* kann oft von Anfang an ein Ausgleich zwischen den Interessen der Beteiligten gefunden werden.

Der Sourcecode eines Computerprogramms ist ebenso wie der Objectcode urheberrechtlich geschützt, wenn er *individuell* ist. Er kann

<sup>149</sup> Spezialisierte Escrowagenten bieten in der Regel verschiedene *Überprüfungsdienstleistungen* an, welche von der Kontrolle, ob die Datenträger elektronisch lesbar sind, bis zur gemeinsamen Prüfung des hinterlegten Sourcecodes mit den Vertragspartnern reichen.

<sup>150</sup> In der Regel wird eine *Herausgabe* für den Konkursfall des Herstellers und bei Leistungsstörung (eventuell nur mit richterlicher Ermächtigung) vorgesehen. Vgl. den Überblick über weitere typische Herausgabebetstände bei *Blickenstorfer* (Fn. 148), S. 240 f.

gegenüber dem Objectcode unter Umständen *zusätzliche Informationen* enthalten, welche eigenen urheberrechtlichen Schutz genießen.

Die Veräußerung eines Computerprogramms führt nicht automatisch zur *Erstveröffentlichung* des Sourcecodes. Auch ein noch nicht veröffentlichter Sourcecode darf jedoch im Rahmen des Weiterveräußerungsrechts von Art. 12 Abs. 2 URG zusammen mit dem Computerprogramm weitergegeben werden.

Jedes Ablaufen, Kopieren oder Ändern eines Computerprogramms tangiert das Urheberrecht. – Bereits beim Laden des Programms entsteht eine temporäre Kopie im Arbeitsspeicher des Computers. Der rechtmäßige Erwerber bzw. Lizenznehmer darf indessen nach Art. 12 Abs. 2 URG analog zu Art. 5 Abs. 1 CRRL alle Handlungen vornehmen, welche zum bestimmungsgemäßen Gebrauch des Computerprogramms notwendig sind. Darunter kann auch die Vornahme notwendiger *Modifikationen* am Sourcecode (insbesondere zur Fehlerbehebung) fallen. Im Ergebnis wird der Umfang des bestimmungsgemäßen Gebrauchs durch eine *Abwägung zwischen den Interessen* des Erwerbers und dem Geheimhaltungs- und Partizipationsinteresse des Urheberrechtsinhabers konkretisiert. Letzteres fällt bei der Fehlerbehebung allerdings außer Betracht. Es ist auch im schweizerischen Recht davon auszugehen, dass das Gebrauchsrecht einen zwingenden Kerngehalt hat.

Das schweizerische URG unterscheidet analog zur CRRL zwischen der den Objectcode nicht verändernden *Beobachtung des Programmablaufs* und der Rekonstruktion des Sourcecodes durch Dekompilierung des Objectcodes. Eine bloße Beobachtung ist zur Analyse eines Computerprogramms allerdings selten ausreichend. Die *Dekompilierung* von Computerprogrammen im Hinblick auf Schnittstellen zur Schaffung interoperabler Programme ist in der Schweiz unter analogen Einschränkungen wie in der EU zulässig. Darüber hinaus kann sich aus dem *Recht zum bestimmungsgemäßen Gebrauch* ausnahmsweise ein Dekompilierungsrecht ergeben, soweit zur Programmverwendung notwendige Informationen anders nicht beschafft werden können.