

Software vs. Copyright

Bitte verteidigt mich nicht mehr

Eine verheißungsvolle Brautzeit

So vernünftig es vielen Leuten auch erscheinen mag, dass Software dem Copyright¹ unterliegt², so handelt es sich dabei in Wirklichkeit doch um eine willkürlich getroffene Entscheidung. Sie war das Ergebnis von Verhandlungen in den 1970er Jahren, die ein regulatorisches Rahmenwerk für Computerprogramme festlegen sollten.

Zur Verteidigung dieser Entscheidung lässt sich anführen, dass ihr eine zutreffende Beobachtung über das Wesen der Programme zugrunde liegt: Es handelt sich um kulturelle Arbeiten, die geschriebenen Werken, nicht aber technisch gefertigten Produkten entsprechen. Diese Beobachtung ist von grundlegender Bedeutung, da damit anerkannt wird, dass Computerprogramme sich für die Kommunikation von Algorithmen eignen, vergleichbar den Partituren, über die Musik kommuniziert wird, oder Gleichungen, die bestimmte mathematische Wahrheiten zum Ausdruck bringen.

Wenn Software als ein Ausdrucksmittel behandelt wird, hat das für die Gesellschaft wichtige und vorteilhafte Konsequenzen. Es bedeutet zum Beispiel, dass das Recht auf freie Meinungsäußerung genauso denjenigen zusteht, die sich in Programmiersprache ausdrücken, wie denjenigen, die Spanisch sprechen oder mit Noten ihre Musik schreiben. Es bedeutet auch, dass Software als Teil unserer Kultur anerkannt wird, deren Ausübung durch die Menschenrechte garantiert

- 1 In diesem Text werde ich von «Copyright» statt «Urheberrecht» sprechen, denn es umfasst die Teilmenge der Urheberrechte, die mit wenigen Ausnahmen überall in der Welt anerkannt sind. Unabhängig von den Details dieser Diskussionen, verändert sich die Argumentation nicht.
- 2 Man kann häufig lesen, dass Software durch das Urheberrecht «geschützt» ist. Ob dieser Schutz wirklich besteht, ist schwer zu beurteilen, weil diejenigen, die dieses Bild verwenden, leider selten dazu sagen, vor welchen Risiken es denn geschützt wird. Zu sagen, dass Software dem Urheberrecht unterworfen ist, entspricht vielmehr der tatsächlichen Situation: Alle Software unterliegt dem Urheberrecht, ungeachtet der Wünsche seiner Autoren und Nutzer.

wird, und dass die Regulierung, der sie unterliegt, ihrer Verbreitung förderlich sein muss, damit sie der Öffentlichkeit zur Verfügung gestellt werden kann.

Eine andere wichtige Konsequenz ist, dass diese Regulierung das Werk (das Programm) und nicht dessen Funktion betrifft: Wenn ich ein Programm schreibe, damit der Computer eine bestimmte Aufgabe erfüllt, kann jemand anderes ein ganz anderes Programm schreiben, das dieselbe Aufgabe erfüllt, ohne dass dies zu juristischen Konflikten führt. Die gegenwärtig in den USA laufende Kontroverse über die Patente auf Software zeigt die Folgen auf, wäre Software fälschlicherweise als Produkt und nicht als Werk behandelt worden: In den USA ist es möglich, ein Patent auf die «Verwendung eines Programms zur Lösung des Problems X» erteilt zu bekommen. Von diesem Moment an, verfügt der Halter des Patentes als Einziger über das Recht, Programme zu schreiben, die Problem X lösen können.³

Eine konfliktreiche Ehe

Wenn Programmieren als Ausdrucksmittel anerkannt ist, ist damit klar, dass Patente nicht der angemessene regulatorische Rahmen für Software sind. Damit ist nicht unbedingt gesagt, dass das Copyright der richtige Rahmen sein muss oder zumindest nicht, dass es vernünftig ist, dasselbe Copyright gleichermaßen auf Programme wie auch auf Bücher oder Lieder anzuwenden.

Beim Urheberrecht ist es ganz klar der Zeitfaktor, der sich als unangemessen für Software erweist. Das Copyright ist ein zeitlich begrenztes Monopol. Es wird davon ausgegangen, dass nach dessen Ablauf das Werk weiterhin einen Nutzen für die Öffentlichkeit hat. Bei der Mehrzahl musikalischer und literarischer Arbeiten können wir tatsächlich davon ausgehen, dass sie noch für lange Zeit von Nutzen sein werden.⁴ Softwareprogramme haben jedoch ein sehr begrenztes Leben. Die rasante Weiterentwicklung der Hardware und ständig neu hinzukommende Anwendungsbereiche führen dazu, dass kaum fünf Jahre nach Veröffentlichung kein Programm ohne Weiterentwicklung nutzbar ist. Ein Programm, das zehn Jahre nach seiner Veröffentlichung der Öffentlichkeit zur Verfügung steht, wäre für die praktische Nutzung bereits unbrauchbar.

Die im Copyright implizierte Idee der gesellschaftlichen Transaktion trifft für Software auch in einem weniger offensichtlichen Aspekt nicht zu. Wenn ein Autor ein Werk unter Copyright veröffentlicht (ein Buch, ein Gemälde, eine musikalische Komposition), präsentiert er dies unmittelbar der Öffentlichkeit.

3 Interessanterweise kann dieses Patent erlangt werden, obgleich der Antragsteller das Programm noch nicht geschrieben hat, geschweige denn die Absicht hegt, dies zu tun.

4 Die gegenwärtig überlange Dauer des Copyrights lässt diese Annahme als übertrieben optimistisch erscheinen: Es ist sehr wahrscheinlich, dass es 70 Jahre nach dem Tod des Autors keine brauchbaren Exemplare des Werkes mehr gibt, sei dies, weil das Medium (Papier, Vinyl, Zelluloid) verrottet ist oder weil das veraltete Format dazu führt, dass es keine Geräte mehr gibt, die darauf Zugriff gewähren. Wie würden wir uns heute etwa Musik anhören, die auf 8-Spur-Kassetten aufgenommen wurde?

Das Publikum kann es studieren, analysieren und alle Aspekte wertschätzen, die das Werk ausmachen. Bei Computerprogrammen geschieht das nicht zwangsläufig: die Programmierer haben die Möglichkeit, das Monopol über ihr Werk auszuüben, ohne es zu veröffentlichen. Dies geht nur deshalb, weil es verschiedene Möglichkeiten gibt, ein Programm darzustellen. Einige dieser Darstellungsformen sind für den Menschen praktisch unverständlich, da sie für die Interpretation durch Computer geschrieben werden. Selbstverständlich benutzen die Menschen, die die Programme schreiben, diese Ausdrucksweise nicht direkt, sondern setzen Programmiersprachen ein, also formelle Notationen, die zum einfachen Verständnis durch Fachleute entwickelt wurden, obgleich sie dem ungeübten Auge wie eine Kreuzung zwischen Englisch und Mathematik vorkommen. In der Sprache «C» könnte beispielsweise ein Programm zur Berechnung der Quadratwurzel einer Zahl folgendermaßen ausgedrückt werden:

```
/* Diese Funktion druckt die Quadratwurzel ihres Arguments */
static void printsqrt(float x) {
  if (x < 0) /* die Wurzel einer negativen Zahl ist imaginär */
    printf(«Die Zahl ist kleiner als null!\n»);
  else /* die Zahl ist positiv, alles in Ordnung */
    printf(«%f\n», sqrt(x));
}
```

An diesem Ausschnitt eines Computerprogramms lässt sich die kommunikative Absicht eines Programms feststellen. Sie wird in einer «menschlichen» Notation ausgedrückt, die erklärende Anmerkungen über die Absicht des Autors enthält sowie die Gründe, weshalb er bestimmte Entscheidungen getroffen hat. Damit kann es von anderen Personen verstanden werden. Diese den Menschen begreifliche Form der Darstellung des Programms wird für gewöhnlich «Quellcode» genannt. Der Computer führt jedoch nicht den Quellcode, sondern das Ergebnis seiner automatisch erfolgten Übersetzung in Computersprache aus. Ein Programm in Computersprache besteht aus einer langen Liste numerisch kodierter Anweisungen, die die grundlegenden, vom Prozessor auszuführenden Operationen auflisten. Wenn wir das obige Programm übersetzen, damit es in Computern des Typs «PC» ausgeführt werden kann, verschwinden alle kommunikativen Elemente, und es wird auf die folgende Zahlenliste reduziert:

2212858197	1171855596	3673086216	2665537513
250282615	1680082119	3892839557	4294967036
1171856363	605871368	4294901736	610065919
604292868	134514050	4294893544	1438894591

Das Problem besteht darin, dass das Urheberrecht nicht nur auf den Quellcode Anwendung findet, sondern auch auf Software in Computersprache, selbst wenn nur letztere verbreitet wird. Aber die Transaktionsidee des Copyright geht davon

aus, dass der Autor ein Monopol auf die Verwertung seines Werkes erlangt, während er dieses gleichzeitig öffentlich zugänglich macht. Wenn ein Programm nur in Computersprache verbreitet wird, kommt es zu keiner Veröffentlichung, und die Öffentlichkeit wird betrogen.

Die Kinder verlangen die Scheidung

Die Free Software Foundation⁵ hat schon viel über die Schäden veröffentlicht, die der Gesellschaft entstehen, wenn der Vertrieb von Software monopolisiert wird und engagiertes Verhalten, wie etwa das solidarische Teilen unter Gleichgesinnten, kriminalisiert wird. Aber darüber hinaus verhindert die Anwendung des Copyright auf Software das Florieren der Computerprogramme als kulturelle Ausdrucksmittel, wenn die sie von anderen Medien unterscheidenden Charakteristika nicht berücksichtigt werden.

Die allgemeine Praxis, Software ohne den Quellcode zu vertreiben (was eher der Verbreitung eines Industriegeheimnisses ähnelt als einem Urheberrecht), erschwert das gegenseitige Lernen, wie es in anderen Künsten praktiziert wird. Es erschwert außerdem die effektive Ausübung des Urheberrechts über jene Werke, die wirklich als Quellcode veröffentlicht wurden: Es ist sehr schwer aufzudecken und nachzuweisen, dass ein Programm ein Plagiat eines anderen enthält, wenn wir nicht über den Quellcode des Originals verfügen, sondern nur über eine Zahlenreihe, innerhalb derer eine der vielen möglichen Übersetzungen des Programms versteckt sein kann oder auch nicht.

Das Ergebnis dieser Verzerrung ist in der Praxis einfach zu erkennen: Großunternehmen schaffen einen Markt, privatisieren für ihren Zweck die Werke ihrer Angestellten und halten das Monopol auf den Vertrieb aufrecht, wobei sie weder das öffentliche Kulturgut bereichern noch nützliche Werke für die Öffentlichkeit schaffen. Gleichzeitig unterliegen sie einem nur minimalen Risiko, bei der Verwendung fremder Software in ihren Programmen entdeckt zu werden. Bleibt die Frage, wie die Softwarelandschaft aussähe, wenn sich in jenen Verhandlungen in den 1970ern die Idee durchgesetzt hätte, dass Software zwar einer Regulierung ähnlich dem Copyright bedarf, diese jedoch an den besonderen Charakter der Software anzupassen sei, mit drastisch verringerten Laufzeiten und der zwingenden Veröffentlichung des Quellcodes.

Auf jeden Fall zeigt die große Zahl an Entwicklern von freier Software, die bewusst darauf verzichten, der Verbreitung, Erforschung und Herstellung von Werken, die sich aus ihren Programmen ableiten, Restriktionen aufzuerlegen, dass das Copyright für die Entwicklung von Software nicht notwendig ist. Es zeigt auch, dass eine auf Kooperation und Zusammenarbeit aufbauende Umgebung sehr viel anregender ist als eine, die sich auf Isolierung und Restriktionen gründet.

5 <http://fsf.org>