

von Rechtsanwältin **Elisabeth Keller-Stoltenhoff**

Rechtlicher Quellcode-Check: Was man über den Software-Quellcode wissen sollte!

Je mehr in unserer alltäglichen Welt die Nutzung von Computerprogrammen unverzichtbar wird, desto mehr scheint es, dass die Kenntnis der Nutzer darüber sinkt, was eigentlich zum Einsatz kommt, wenn ein Handy, ein Computer oder eine Waschmaschine genutzt werden. Allen diesen Geräten ist gemein, dass sie von Softwareprogrammen gesteuert werden.

Wie aber entsteht diese Software? Welche Rechte hat der Nutzer an dieser Software und welche nicht? Worauf sollte er achten, wenn er die Software für sich erstellen lässt? Bekommt er nach der Erstellung auch den Quellcode oder nur den Maschinencode? Was ist überhaupt ein Quell- oder ein Maschinencode?

Die IT-Recht-Kanzlei wird mit solchen und ähnlichen Fragen tagtäglich konfrontiert. Der folgende Beitrag soll Antworten auf die häufigsten Fragen bieten.

1. Was ist der Unterschied zwischen Quellcode und Maschinencode?

Bei dem Maschinencode, der auch oft Objectcode genannt wird, handelt es sich um eine für einen Menschen kaum verstehbare Abfolge von Nullen und Einsen, dem so genannte Binärcode. Das liegt daran, dass Geräte nur zwei gegensätzliche Zustände erkennen können, zum Beispiel An/Aus, Wahr/Falsch, 0/5 Volt im Stromkreislauf oder - eben beim Binärcode - 1 oder 0. Die Basiseinheit des Binärcodes bildet das Bit, eine Speicherstelle, welche nur den Wert 0 oder 1 annehmen kann. Acht solcher Bits bilden ein Byte, mit dem sich dann 256 Zeichen darstellen lassen.

Nur wenige Menschen können Handlungsbefehle an eine Maschine direkt in der Maschinensprache schreiben. Z.B. versteht kaum jemand, was hiermit gemeint sein könnte:

■ `"0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000"`

Programmierer greifen daher in der Regel auf so genannte Programmiersprachen zurück, um ihre Anweisungen für den Computer zu schreiben. Die mit diesen Sprachen erstellten Programme werden Quellcode oder auch Sourcecode genannt. Programmiersprachen wie Cobol, C++, Java oder Perl haben für Menschen verständliche Begriffe und zum Teil auch sehr umfangreiche Werkzeuge, die mit einem Befehl genutzt werden können. Ein Beispiel für einen Quellcode ist der folgende Ausschnitt eines

C-Quellcodes:

```
// C Code
bool isPrime(int nNum) {
int maxFactor = sqrt(nNum), thisFactor;
bool bRetVal = true;
for(int i = 0; i < list_size; ++i) {
thisFactor = prime_number_list[i];
if(thisFactor > maxFactor) {
break;
}
if(nNum % thisFactor == 0) {
bRetVal = false;
break;
}
}
return bRetVal;
}
```

Da hier nicht nur Nullen und Einser vorkommen, steht fest, dass diese Programmzeilen des Quellcodes erst in Maschinsprache übersetzt werden müssen, bevor sie von einem Computer ausgeführt werden können. Diese Übersetzungsarbeit erledigt wiederum ein Programm, der so genannte Compiler.

Nach der Übersetzung liegt dann das Programm als binärer Code vor, der zwar vom Computer ausgeführt aber nicht mehr verändert werden kann. Modifiziert werden kann dieses Programm nur über eine Änderung des zugrunde liegenden Quellcodes und eine erneute Übersetzung in den Maschinencode durch den Compiler.

Fazit: Will ein Nutzer ein Softwareprogramm verändern können, muss er also sowohl über dessen Quellcode als auch den passenden Compiler verfügen.

2. Welche Rechte erhält der Nutzer am Quellcode und am Sourcecode?

Der Objectcode einer Software und der dazugehörige Quellcode gehören zusammen. Meist werden aber Nutzungsrechte nicht für beide Programmtypen gewährt. Programme die unter einer [Open Source Lizenz](open-source-software-nutzungsrechte.html?search=open+source) stehen, werden in der Regel direkt mit ihrem Quellcode ausgeliefert, dessen Änderung und Weitergabe durch die Lizenz gestattet ist. Dies ermöglicht deren Anpassung für spezifische Bedürfnisse.

Proprietäre Software wird meist ohne Quellcode ausgeliefert. Die Rechte des Nutzers beziehen sich dabei lediglich auf den Maschinen- oder Objectcode. Dem Schutz eines Abnehmers dienen dann manchmal Vereinbarungen zur Quellcodehinterlegung (Source Code Escrow Agreements).

Ein Nutzer eines Standardprogramms wie z.B. Word wird aber auch in der Regel kaum an einer Änderung des Programms und daher an dem Quellcode interessiert sein. Relevant wird die Frage der Überlassung des Quellcodes und der Rechte an diesem meist erst bei der Erstellung von Individualsoftware. Denn in solchen Fällen ist der Auftraggeber oft durchaus daran interessiert, die Software auch ohne Hilfe des Auftragnehmers ändern und ohne dessen Zustimmung verwerten zu können.

Nach der **Zweckübertragungsregel** des Urheberrechts wird aber im Zweifel zu Lasten des Rechteeerwerbers angenommen, dass der Auftragnehmer als Urheber ein Nutzungsrecht nur in demjenigen Umfang einräumen will, den der Vertragszweck unbedingt erfordert. Anzuknüpfen ist dabei an die nächstliegende Verwertungsform, und zwar auch dann, wenn sich der Geschäftsbetrieb des Verwerters offensichtlich auch auf andere Verwertungsbereiche erstreckt.

Die vertraglich eingeräumten Nutzungsrechte werden sich daher in der Regel nicht auch auf den Quellcode beziehen, es sei denn die Parteien haben ausdrücklich vereinbart, dass der Auftraggeber auch berechtigt sein soll, die Software zu verändern. Trifft letzteres zu, dann umfasst die bestimmungsgemäße Nutzung auch die Herausgabe des Quellcodes und das Recht, diesen zu verändern.

Tipp: Aber auch wenn umfangreiche Rechte am Quellcode der Individualsoftware bestehen, sollte der Auftraggeber eines Softwareerstellungsvertrages folgende Regelungsnotwendigkeiten nicht außer Acht lassen:

- Wichtig ist der Zeitpunkt des Übergangs der Rechte. Auftragnehmer bevorzugen hier den Zeitpunkt der Zahlung oder frühestens der Lieferung bzw. der Abnahme. Auftraggeber hingegen haben Interesse daran, diesen Zeitpunkt auf den Moment der jeweiligen Erstellung der Software vorzuverlegen, um im Insolvenzfall des Auftragnehmers vor der Abnahme nicht den Anspruch auf **Übertragung der Nutzungsrechte** zu verlieren, wenn der Insolvenzverwalter gem. § 103 Abs. 1 InsO die Erfüllung des Vertrages ablehnt, weil der Vertrag noch nicht vollständig erfüllt ist.
- Wichtig ist auch, vertraglich zu vereinbaren, wie umfangreich der Quellcode vom Auftragnehmer zu kommentieren ist. Die Logik in einem unkommentierten Quellcode kann oft selbst von seinem Verfasser nach einiger Zeit kaum noch nachvollzogen werden. Hier hilft nur, dass der Verfasser jeden Schritt des Quellcode kommentiert und darstellt, warum er hier welche Kodierung vornimmt. Nur so kann sich ein Dritter in dem Quelltext zurecht finden und ihn ggf. ändern.
- Wie oben dargestellt, nutzt ein Quellcode nichts, wenn nicht auch ein Übersetzungsprogramm (Compiler) vorliegt, das den Quellcode in den Maschinencode übersetzt. Ist der Compiler auf dem Markt erhältlich, ist das kein Problem für den Auftraggeber. Problematisch wird die Angelegenheit aber dann, wenn der Auftragnehmer einen selbst erstellten oder aus anderen Gründen nicht frei verfügbaren Compiler nutzt. Dann müssen die vertraglichen Regelungen zwingend auch die Überlassung dieses Compilers umfassen, wenn der Auftraggeber die Software bearbeiten oder ändern will.

Fazit

Quellcode und Maschinencode sind zwei unterschiedliche Teile eines Computerprogramms. Bei Standardsoftware wird der Quellcode in der Regel nicht überlassen, es sei denn, es handelt sich um Open-source-Software. Bei Individualsoftware sollte das Bearbeitungsrecht der Software und die Übergabe des Quellcodes stets detailliert geregelt werden, da der Auftraggeber ohne eine solche Regelung oft weder ein Bearbeitungsrecht noch das Recht auf Überlassung des Quellcodes erhält.

Autor:

RAin Elisabeth Keller-Stoltenhoff

Rechtsanwältin