

Seminarvortrag

Recht für Software-Entwickler

Anne Pilchowski Volker Grabsch

12. Juni 2008

Veröffentlicht unter

<http://www.prof.v.de/uni/>

Lizenziert unter



Creative Commons BY-SA 3.0

Inhaltsverzeichnis

1 Was ist Recht?	3
2 Arbeitsrecht	4
2.1 Nicht-Selbstständige	4
2.2 Selbstständige	4
3 Kaufrecht	6
3.1 Kaufvertrag	6
3.2 Werkvertrag	7
3.3 Wichtige Punkte im Werkvertrag	7
4 Beispiel: DC-Rap	9
5 Urheberrecht	12
5.1 Einführung	12
5.2 Software-Lizenzen	13
6 Fortsetzung: DC-Rap	16
7 Schlussfolgerungen	17
Literatur	18

1 Was ist Recht?

Das Wort Recht ist ein altes gemein-germanisches Wort. [Wö07] Im objektiven Sinn, also die Gemeinschaft betreffend, ist es das geregelte menschliche Zusammenleben sowie das soziale Miteinander, das durch staatlichen Zwang garantiert wird. Subjektiv (eine einzelne Person betreffend) sind es die Befugnisse einer Person, wie sie aus dem Objektiven abgeleitet werden können, die sie befähigen etwas zu tun oder zu unterlassen. [Wik08f]

Einfach ausgedrückt ist Recht die Gesamtheit der Normen, das heißt alle Befugnisse, Pflichten und Verbote, die die Mitglieder einer Gesellschaft haben. Es dient dazu, Ordnung zu schaffen und Konflikte, die im Zusammenleben auftreten, zu lösen. Während die Moral bzw. Sitte mit dem gleichen Ziel auf die innere Gesinnung der Menschen Einfluss nimmt, regelt das Recht die äußeren Handlungen und zieht bei Nichteinhaltung staatliche Sanktionen nach sich.

Um in praktischen Fällen das geltende Recht zu ermitteln, wird auf Rechtsquellen (zum Beispiel das Bürgerliche Gesetzbuch [BGB08]) zurückgegriffen.

Für die Gliederung aller Unterbereiche des Rechts gibt es viele Möglichkeiten. Eine davon ist die Unterscheidung zwischen nationalem Recht und Völkerrecht. Ersteres beinhaltet das Bundes- und das Landesrecht, das jeweils in ganz Deutschland oder nur in den einzelnen Bundesländern gilt. Das zweite setzt sich aus den Staatsverträgen zusammen, die zwischen mindestens 2 Staaten geschlossen wurden.

Gebräuchlich ist auch die Einteilung in öffentliches und Privatrecht. Das öffentliche Recht regelt das Verhältnis zwischen Privatpersonen (natürlichen Personen und Personengesellschaften) und den Trägern öffentlicher Gewalt. Das Bürgerliche Recht oder Zivilrecht, wie das Privatrecht auch genannt wird, reguliert die Beziehung von rechtlich gleichgestellten Personen.

Für Softwareentwickler besonders interessant sind die drei Bereiche Arbeitsrecht [BGB08, §611 – §630], Kaufrecht [BGB08, §433 – §480, §631 – §651] und Urheberrecht [Urh08]. Das erste legt fest, unter welchen Bedingungen der Programmierer arbeiten soll und wie er dafür entlohnt wird. Das zweite ist für die Verträge zwischen Softwarefirma und Käufer zuständig und letzteres regelt die Befugnisse der Programmierer, andere Software oder Softwarefragmente zu nutzen. [Ott06]

2 Arbeitsrecht

Damit eindeutig festgelegt ist, wie ein Programmierer zu arbeiten hat und auf welche Art er dafür bezahlt wird, werden, genau wie zwischen anderen Arbeitgebern und Arbeitnehmern, Verträge aufgesetzt, in denen die Pflichten und Rechte beider Parteien festgehalten werden. Dabei kann der Software-Entwickler entweder als Selbständiger oder als Nicht-Selbständiger für eine Firma programmieren.

2.1 Nicht-Selbstständige

Nicht-Selbstständige sind Angestellte oder Arbeiter. Rechtlich gibt es heutzutage in Deutschland keine Unterscheidung mehr zwischen diesen beiden Formen. Allerdings erhält der Angestellte ein sogenanntes Gehalt, also einen monatlich festgesetzten Betrag, während der Arbeiter Lohn bekommt, der auf der Stundenanzahl seiner geleisteten Arbeit basierend jeden Monat schwanken kann. Im Gegensatz zu selbständigen Personen sind die Nicht-Selbstständigen vertraglich an das Unternehmen gebunden. [Wik08a] Außerdem übernimmt der Arbeitgeber einen Teil der Nebenkosten und Versicherungsbeiträge: „Bei versicherungspflichtig Beschäftigten ... trägt der Arbeitgeber die Hälfte der Beiträge des Mitglieds aus dem Arbeitsentgelt nach dem um 0,9 Beitragssatzpunkte verminderten allgemeinen Beitragssatz; im Übrigen tragen die Beschäftigten die Beiträge.“ [SGB08, §249(1)]

2.2 Selbstständige

Der freie Mitarbeiter wird in der IT-Branche auch Freelancer genannt. Er führt in Unternehmen Aufträge aus, ohne dort eingegliedert zu sein und wird entweder stundenweise (im Dienstvertrag festgelegt) oder pauschal (mit Werkvertrag) bezahlt. Er ist meist hochqualifiziert und spezialisiert und wird von einem Unternehmen eingesetzt, wenn dieses für spezielle Aufträge zu wenig eigene Angestellte hat, sich das dauerhafte Einstellen von Festangestellten aber nicht lohnt. Nebenkosten, Versicherungen etc. muss er von seinem Lohn selbst bezahlen und aufgrund der zeitlichen Begrenzung seines Vertrages hat er auch kein gesichertes Einkommen, sondern muss sich ständig um neue Aufträge bemühen. [Wik08b]

Andererseits ist der Selbstständige in der Gestaltung seiner Arbeitsbedingungen frei, das heißt, er kann arbeiten, wann und wo er möchte. Er ist weder in zeitlicher oder örtlicher noch in fachlicher Hinsicht den Weisungen des Auftraggebers direkt unterworfen. „Zu der freiberuflichen Tätigkeit gehören die selbständig ausgeübte wissenschaftliche, künstlerische, ... Tätigkeit, die selbständige Berufstätigkeit der ..., Ingenieure, ... und ähnlicher Berufe.“ [ESt08, §18(1)1.]

Eine weitere Form eines Selbstständigen ist der Subunternehmer. Er wird ebenfalls an einem Auftrag eines Unternehmens beteiligt, ohne zum Unternehmen zu gehören. Allerdings bedarf es einer Genehmigung des Auftraggebers, dass er sich an der Arbeit beteiligen darf, während das Unternehmen in der Wahl seiner freien Mitarbeiter unabhängig ist.

„Ein Angehöriger eines freien Berufs ... ist auch dann freiberuflich tätig, wenn er sich der Mithilfe fachlich vorgebildeter Arbeitskräfte bedient; Voraussetzung ist, dass er auf Grund eigener Fachkenntnisse leitend und eigenverantwortlich tätig wird. Eine Vertretung im Fall vorübergehender Verhinderung steht der Annahme einer leitenden und eigenverantwortlichen Tätigkeit nicht entgegen.“ [ESt08, §18(1)1.]

3 Kaufrecht

Ein weiteres für Programmierer wichtiges Rechtsgebiet ist das Kaufrecht. In diesem werden die Rechte und Pflichten zwischen Verkäufer, also der Software-Firma, und dem Käufer der Software geregelt. Man unterscheidet zwischen 2 Arten von Kaufvertrag: dem eigentlichen Kaufvertrag für Standardsoftware und einem Werksvertrag, der dann eingesetzt wird, wenn individuelle Software programmiert wird. [VDI06]

3.1 Kaufvertrag

Wird Standard-Software produziert, also Software, die jeder käuflich erwerben kann, ohne dass noch einmal individuelle Anpassungen vorgenommen werden, dann wird ein normaler Kaufvertrag zwischen Käufer und Firma geschlossen, wie er auch bei anderen Handelsgütern üblich ist. [Wik08d]

Dieser beinhaltet üblicherweise, dass der Verkäufer dem Käufer die Vertragsache mängelfrei (d.h. frei von Sach- und Rechtsmängeln) zu übergeben hat und dafür von diesem den vereinbarten Kaufpreis erhält. [BGB08, §433]

Es wird in aller Regel nicht die Software an sich verkauft, sondern eine Lizenz, die den Käufer dazu berechtigt, das Programm zu nutzen. Außerdem gibt es meist auch zusätzliche Leistungen, die beim Kauf erworben werden. Wenn die Software nicht gerade heruntergeladen wird, ist im Preis auch das Speichermedium, auf dem sie gespeichert ist, inbegriffen. Handbücher und nachträgliche Support-Leistungen können ebenfalls dazugehören.

Sollte das Produkt nicht den Erwartungen des Käufers entsprechen, gibt es noch Gewährleistungen beziehungsweise Garantien.

Die Gewährleistungen ist eine gesetzlich vorgeschriebene Leistung. Sie stellt den Anspruch des Käufers auf Nachbesserung sicher, falls die Sache mangelhaft geliefert wird. Dabei handelt es sich um einen Sachmangel, wenn sie bei Gefahrübergang nicht die vereinbarte Beschaffenheit hat. [BGB08, §434(1)], oder um einen Rechtsmangel, wenn Dritte einige nicht im Kaufvertrag übernommene Rechte gegen den Käufer geltend machen können. [BGB08, §435]

Unter dem juristischen Begriff „Gefahrübergang“ versteht man vereinfacht gesagt den Zeitpunkt der Übergabe der Sache vom Verkäufer an den Käufer bzw. des Verkäufers an eine Spedition zur Lieferung an den Käufer. [lex07a]

Die Gewährleistung gilt mindestens 2 Jahre, allerdings liegt die Beweislast nur die ersten 6 Monate beim Verkäufer. [BGB08, §476] Danach muss der unzufriedene Kunde nachweisen, dass der Fehler nicht durch ihn entstanden ist.

Eine weitere Schadensersatzleistung ist die Garantie. Diese wird freiwillig zuge-

sichert und kann deshalb beliebige Klauseln und Einschränkungen enthalten. Auch in der einschlägigen Werbung gegebene Garantien sind für den Verkäufer bindend. [BGB08, §443]

3.2 Werkvertrag

Muss die gewünschte Software erst programmiert oder angepasst werden, dann wird zwischen der Firma und dem Kunden ein Werkvertrag abgeschlossen. „Durch den Werkvertrag wird der Unternehmer zur Herstellung des versprochenen Werkes, der Besteller zur Entrichtung der vereinbarten Vergütung verpflichtet. Gegenstand des Werkvertrags kann sowohl die Herstellung oder Veränderung einer Sache als auch ein anderer durch Arbeit oder Dienstleistung herbeizuführender Erfolg sein.“ [BGB08, §631]

Daher stehen im Werkvertrag die genauen Anforderungen drin. Werbeaussagen, die der Programmierer macht, sind ebenfalls verbindlich und müssen erfüllt werden. Der Käufer ist verpflichtet, gleich nach dem Erhalten des Produktes dieses im Rahmen einer Abnahme auf Mängel zu prüfen. Erfolgt die Abnahme nicht fristgemäß, so kann sie nicht nachgeholt werden, sondern gilt als bestanden: „Der Abnahme steht es gleich, wenn der Besteller das Werk nicht innerhalb einer ihm vom Unternehmer bestimmten angemessenen Frist abnimmt, obwohl er dazu verpflichtet ist.“ [BGB08, §640(1)]

Unwesentliche Mängel dürfen kein Grund sein, die Abnahme zu verweigern. [BGB08, §640(1)] Entscheidend ist nur, dass die Software die im Vertrag festgelegten Anforderungen erfüllt.

3.3 Wichtige Punkte im Werkvertrag

Um Probleme und Streitigkeiten von vornherein zu verhindern, sollte beim Erstellen eines Werkvertrages auf einige wichtige Punkte geachtet werden:

Der Auftraggeber sollte das konkrete Konzept und ein Pflichtenheft schriftlich festhalten. Ebenso muss der Preis, der Termin der Fertigstellung und der Zeitraum zur Prüfung der Software vorher vereinbart werden. Im Vertrag sollte auch stehen, dass der Kunde das fertige Produkt nutzen und bearbeiten darf. Dafür ist es hilfreich, wenn er den Quellcode und eine zugehörige Dokumentation erhält, was ebenfalls vertraglich festgehalten werden muss. Ein weiterführender Support wie Wartung und Pflege der Software kann ebenfalls bei Bedarf angefordert werden. Besonders wichtig für den Käufer ist, dass er sich vor Vertragsabschluss über das Insolvenzrisiko der Firma informiert und prüft, ob die Vermögensschadenshaftpflichtversicherung

ausreichend ist. Er sollte außerdem keine Vereinbarungen mit anderen Parteien treffen, was die weitere Nutzung der gewünschten Software betrifft, bevor der Vertrag unterzeichnet ist.

Dem Auftragnehmer wird empfohlen, dass er für jede Projektphase einen gesonderten Vertrag erstellt. Da die genaue Einschätzung des Aufwandes sehr schwer oder sogar unmöglich ist, muss er sich zusätzlich absichern. So muss der Vorbehalt der technischen Machbarkeit im Vertrag aufgenommen werden. Ebenso sollte er zusätzliche Aufwendungen genehmigen lassen und, falls der Auftraggeber doch noch Änderungswünsche hat, ein formelles Verfahren für Change Requests vereinbaren. Damit Terminengpässe rechtzeitig erkannt werden, sollte die Firma den Auftraggeber fortwährend schriftlich über Fortschritte und Probleme informieren.

4 Beispiel: DC-Rap

DC-Rap ist eine Software, die in klinischen Studien für den technischen Import von medizinischen Bilddaten eingesetzt wird. [Gra08] Sie wurde von der „Bayer Schering Pharma AG“ in Auftrag gegeben und durch die Firma „Rent a PACS GmbH – Multimediale Archive“ realisiert. An dem Projekt arbeiteten zwei Programmierer, der eine als freier Mitarbeiter (Selbstständiger) und der andere als Werkstudent (Angestellter).

Der Werkvertrag wurde durch den Auftraggeber Schering vorgegeben und ist ein Standardvertrag für derartige Projekte. Einige Formulierungen im Vertrag erweckten den Eindruck, dass er eher auf Arbeitsergebnisse von Werkstudenten bei Schering vorort ausgerichtet ist. Doch insgesamt passte der Vertrag auch sehr gut auf das Softwareentwicklungs-Projekt.

Der Vertrag stützte sich auf zwei zusätzliche Dokumente, das Lastenheft und das Pflichtenheft. Das Lastenheft wurde vom Auftraggeber entworfen und enthält neben einer groben Beschreibung, was die Software leisten soll, auch interessante Hintergrundinformationen über ihr Einsatzgebiet. Auf dieser Grundlage wurde dann vom Auftragnehmer das Pflichtenheft erstellt, das detailliert aufführt, wie die Anforderungen des Lastenheftes in der Software umgesetzt werden sollen.

Die Erstellung des Pflichtenhefts dauerte einige Monate, doch die Entwicklungsarbeiten begannen sofort. So konnten schon frühzeitig wertvolle technische Erfahrungen gesammelt werden, die z.T. direkt in das Pflichtenheft einfließen. Für den Auftragnehmer war dieses Vorgehen einerseits riskant, da ihm schon vor Vertragsabschluss erhebliche Entwicklungskosten entstanden. Andererseits wurde auf diese Weise sicher gestellt, dass kein Punkt des Pflichtenhefts zu unerwarteten technischen Schwierigkeiten führte.

Dennoch verlief DC-Rap nicht ganz im Zeitplan, die Auslieferung der Version 1.0 kam zwei Monate später als vereinbart. Dies hatte jedoch keine negativen Konsequenzen, da bereits gut funktionierende Versionen zu Verfügung standen. Zudem waren andere Software-Zulieferer viel stärker im Verzug.

Jul 2006	Start der Entwicklung
Okt 2006	offizielle Auftragserteilung
Okt 2006	DC-Rap 0.1
Nov 2006	DC-Rap 0.2
Jan 2007	Termin zur Fertigstellung
...	
Feb 2007	DC-Rap 0.5
Mär 2007	DC-Rap 1.0
...	
Feb 2008	DC-Rap 1.6
Mai 2008	DC-Rap 2.0

Der Auftraggeber erhielt direkt nach der offiziellen Auftragserteilung (Unterzeichnung des Vertrages) eine erste Vorab-Version der Software, um sich damit vertraut zu machen und Änderungswünsche möglichst früh zu äußern. Diese wurden gemeinsam diskutiert und führten zu entsprechenden Änderungen in der Software sowie weiteren spontanen Vorab-Versionen. Dies war vor allem deshalb möglich, weil es auf Seite des Auftraggebers einen festen Ansprechpartner mit allen Entscheidungsbefugnissen gab. Auf formelle Change-Requests wurde bewusst verzichtet.

Ein wesentlicher Teil der Software ist ein Parser für DICOM-Dateien. Die Daten wurden von den unterschiedlichsten medizinischen Geräten erzeugt, die den Standard z.T. unterschiedlich auslegen, oder sich nach inzwischen veralteten Versionen des Standards richten. Hinzu kommt, dass einige Daten mit fehlerhaft arbeitenden Anonymisierungs-Werkzeugen vorverarbeitet wurden. Es war unmöglich, alle Abweichungen vom DICOM-Standard im Voraus zu erahnen. Im Laufe des Projektes gab es immer wieder Überraschungen. Der Auftraggeber testete die aktuellen Entwicklungs-Versionen von DC-Rap immer wieder mit neue Bilddaten. Problemfälle lieferte er sofort an den Auftragnehmer, wo sie analysiert und der Parser entsprechend verbessert wurde.

Der Vertrag sah als Vergütung einen Festbetrag vor, der in zwei Teilen zu zahlen war. Der erste Teil war nach Lieferung der fertigen Software zu zahlen, der zweite Teil nach erfolgreicher Abnahme.

Mit der Zeit kamen auch Wünsche für neue Funktionalitäten, die nicht im Pflichtenheft standen, für die es aber auf Seite des Auftraggebers großen Bedarf gab. Der Auftragnehmer akzeptierte und implementierte diese Wünsche, was jedoch dazu führte, dass der vertraglich vereinbarte Festbetrag schnell aufgebraucht war. Hier kam der Auftraggeber entgegen, indem er für die zusätzliche Funktionalität einen Folge-Auftrag formulierte, mit neuem Pflichtenheft und entsprechender Bezahlung.

Die Zukunft wird zeigen, ob es nach Abarbeitung des zweiten Pflichtenhefts noch einen dritten Vertrag geben wird, für die Wartung der Software.

Der Werkvertrag sah eine Gewährleistung über einen Zeitraum von 24 Monaten vor. Dieser begann jedoch nicht mit der Auslieferung der Software, sondern erst mit ihrer produktiven Nutzung. Weiterhin zählten dabei keine Ausfallzeiten hinzu, die durch Mängel in der Software verursacht wurden. Bisher gab es im produktiven Einsatz von DC-Rap noch keine Ausfälle.

5 Urheberrecht

5.1 Einführung

Das Urheberrecht schützt literarische, wissenschaftliche und künstlerische Werke. [Urh08, §1] Es hat sich zu Zeiten des Buchdrucks entwickelt, gilt aber heute genauso für alle anderen Sprachwerke, Musik, pantomimische Werke, Werke der bildenden Künste, Lichtbildwerke, Filmwerke und wissenschaftliche sowie technische Darstellungen. Computerprogramme sind ebenfalls urheberrechtlich geschützt, sie zählen juristisch zu den Sprachwerken, ebenso wie Schriftwerke und Reden. [Urh08, §2(1)] [Urh08, §69a(4)]

Ein Werk genießt diesen Schutz sofort ab dem Zeitpunkt seiner Erschaffung. Dazu benötigt es weder eine Anmeldung noch irgendeine Markierung. Es muss lediglich eine gewisse Gestaltungshöhe (Schöpfungshöhe) erreicht haben. Wichtig ist auch, dass es sich um ein persönliches Werk der Phantasie handelt, und nicht beispielsweise um eine zwangsläufige wissenschaftliche Erkenntnis. [lex07b]

Jede nichttriviale Software untersteht damit automatisch dem Urheberrecht. Der Schöpfer eines Werkes wird Urheber genannt. [Urh08, §7] Er kann seine Rechte teilweise an andere abtreten, auch exklusiv, doch das Urheberrecht an sich ist nicht übertragbar, außer durch den Tod des Urhebers. In diesem Fall wird das Urheberrecht vererbt. [Urh08, §29]

Das Urheberrecht eines Werkes erlischt erst 70 Jahre nach dem Tod des Autors. [Urh08, §64] Werke, für die das Urheberrecht erloschen ist, oder die von vornherein keine Schöpfungshöhe erreicht haben, nennt man gemeinfrei. [Wik08c]

Unter bestimmten Umständen hat der Urheber schon zum Zeitpunkt der Erschaffung nicht alle Rechte an seinem Werk. „Wird ein Computerprogramm von einem Arbeitnehmer in Wahrnehmung seiner Aufgaben oder nach den Anweisungen seines Arbeitgebers geschaffen, so ist ausschließlich der Arbeitgeber zur Ausübung aller vermögensrechtlichen Befugnisse an dem Computerprogramm berechtigt, sofern nichts anderes vereinbart ist.“ [Urh08, §69b(1)]

Das Urheberrecht ist dem Copyright der angelsächsischen Rechtsprechung sehr ähnlich, auch wenn es feine Unterschiede gibt.

Klar abzugrenzen ist es jedoch vom Patentrecht. Im Gegensatz zum Patentrecht verbietet das Urheberrecht nicht den Nachbau einer Software. Es ist sogar üblich, dass es für ein und den selben Zweck verschiedene Programme gibt, die sehr ähnlich aussehen und funktionieren, aber in Qualität, Funktionsumfang oder Lizenzdetails voneinander abweichen. Der Nachbau einer patentierten Erfindung ist hingegen verboten.

5.2 Software-Lizenzen

Möchte der Urheber anderen erlauben, sein Werk zu nutzen, so geschieht dies durch einen sogenannten Lizenz-Vertrag.

Solch eine Lizenz kann der Urheber anderen Personen individuell einräumen, meist gegen Bezahlung, oder er kann die Lizenz pauschal jedem anbieten. Beides zugleich ist ebenfalls möglich. So ist es gängige Praxis, eine restriktive Lizenz für jedermann, und eine Lizenz mit mehr Freiräumen für die zahlenden Kunden auszustellen.

Bei einer Software hat der Urheber nicht nur Rechte an seinem Quellcode, sondern auch am daraus erzeugten Maschinencode (Objektcode). Daher kann eine Lizenz beispielsweise Rechte am Maschinencode einräumen, aber jegliche Nutzung des Quellcodes verbieten. Eine Lizenz kann jedoch auch sehr weitreichende Rechte einräumen, zum Beispiel jedermann die Ausführung, Anpassung und Weiterverbreitung der Software erlauben.

Je nachdem, wie viele Freiräume eine Lizenz einräumt, nennt man sie *frei* oder *proprietär*. [Gra07] Diese Kategorisierung ist völlig unabhängig davon, ob man die Lizenz kostenlos erhält oder nicht. Ein beliebter Merkspruch ist, dass hier „frei wie in Freiheit“ gemeint ist, nicht „frei wie in Freibier“. [Fre07] Klar abzugrenzen ist Freie Software daher von sogenannter Freeware, die zwar kostenlos erhältlich ist, aber nur unter einer proprietären Lizenz.

Der Begriff der Freien Software geht auf Richard Stallman zurück, der das GNU-Projekt und die Free Software Foundation ins Leben rief. Dort wird eine Software als frei angesehen, wenn jeder Anwender folgende vier Freiheiten erhält: [Fre07]

- die Freiheit, die Software zu starten, egal zu welchem Zweck
- die Freiheit, die Arbeitsweise der Software zu studieren und an die eigenen Bedürfnisse anzupassen
- die Freiheit, das Programm an andere weiterzugeben, kostenlos oder gegen Bezahlung (für den Datenträger, Arbeitsaufwand, etc.)
- die Freiheit, das Programm zu verbessern und die verbesserte Version wieder zu veröffentlichen

Grundvoraussetzung für diese Freiheiten ist der Zugang zum Quellcode. Deshalb hat sich zudem der Begriff der quelloffenen Software (Open Source) etabliert.

Beide Begriffe sind im Prinzip Synonyme, haben aber ideologische Unterschiede. Open Source stellt die wirtschaftlichen Vorteile in den Vordergrund, Freie Software die Freiheit der Anwender. Neuerdings hat sich ein drittes Synonym etabliert, das sich um politische Korrektheit bemüht: FOSS – Freie und Open Source Software.

Damit sich nicht jeder Entwickler Freier Software einen eigenen Lizenztext erarbeiten muss, haben sich einige Standard-Lizenztexte etabliert. Zu den beliebtesten zählen die GNU GPL¹ und die BSD-Lizenz², doch es gibt viele weitere.

Nach dem Vorbild der Freien Software haben sich auch freie Lizenzen für Texte, Bilder, Musik und Filme entwickelt. Zu den bekanntesten gehören die GNU FDL³ und die Creative-Commons-Lizenzen, wobei von letzteren nur die Varianten BY und BY-SA zu den freien Lizenzen zählen.

Das Debian-Projekt ist eine Software-Distribution, die besonders großen Wert darauf legt, nur freie Software, freie Handbuch-Texte, freie Grafiken, etc. zu verwenden. Das Projekt muss daher für sehr viele Lizenzen die Entscheidung treffen, ob sie frei oder proprietär sind. Dabei hat sich herausgestellt, dass die vier Anwenderfreiheiten der FSF nicht genau genug formuliert sind, um als Maßstab für die Bewertung juristischer Lizenztexte zu fungieren. So entstanden die Debian-Richtlinien für Freie Software (DFSG⁴), eine Aufzählung von 10 Punkten, die detailliert beschreiben, wann eine Lizenz frei ist und wann nicht. Die DFSG haben sich so gut etabliert, dass sie fast unverändert von der OSI⁵ als „Open Source“-Definition (OSD) übernommen wurden.

Als proprietär gilt jede Lizenz, die in mindestens einem Punkt der OSD widerspricht. Meist liegen proprietäre Software-Lizenzen entweder sehr dicht an der OSD, oder sie sind kilometerweit davon entfernt. Letzteres bedeutet, dass der Anwender bestenfalls ein einfaches Nutzungsrecht erhält. Außer durch Konfigurationseinstellungen gibt es keine Möglichkeiten zur Anpassung der Software. Selbstständige Fehlerkorrekturen sind ebenso verboten wie die Beauftragung einer anderen Firma mit dem Einbau von Sonderwünschen.⁶

Entsprechend wird erst gar kein Quellcode mitgeliefert. Wenn überhaupt ein Blick auf den Quellcode gestattet ist, dann nur unter Bedingungen, die es letztlich unmöglich machen, das dabei gewonnene Wissen anzuwenden. So kommt es, dass der Anwender höchstens ein Reverse Engineering des Maschinencodes vornehmen kann, doch auch das wird in der Lizenz verboten.

Manche Lizenzen gehen sogar so weit, dass sie vorschreiben, die Software dürfe nur auf einem bestimmten Computer gestartet werden (sog. OEM-Lizenzen). Hier überschreitet der Urheber seine Rechte, solche Bestimmungen im Lizenzvertrag sind

¹GNU General Public License

²Berkeley Software Distribution License

³GNU Free Documentation License

⁴Debian Free Software Guidelines

⁵Open Source Initiative

⁶Das ist vor allem bei älterer Software ein Problem, die vom ursprünglichen Hersteller nicht mehr weiterentwickelt wird.

nach deutschem Urheberrecht ungültig. Dennoch kann er sie durch Kombination von anderen Einschränkungen durchsetzen. Hierzu baut er entsprechende Sperren direkt in die Software ein, die nicht auf legalem Weg umgangen werden können, sondern nur durch Reverse Engineering oder durch Knacken eines Kopierschutzes.

Am anderen Ende der Skala liegen proprietäre Lizenzen, die von der OSD nur geringfügig abweichen, sogenannte halbfreien Lizenzen. Es gibt verschiedene Gründe, aus denen eine Lizenz nur halbfrei ist:

Der Autor von `ircII` verwendete *ausversehen* eine unfreie Lizenz. Die Lizenz erlaubte zwar Änderungen an der Software auf dem eigenen Rechner, enthielt aber keine explizite Erlaubnis, die Änderungen zu veröffentlichen. Als der Autor von dem Problem in Kenntnis gesetzt wurde, wechselte er einfach zur BSD-Lizenz. Seitdem ist `ircII` freie Software. [Sch06]

Der Autor von `Qmail` hatte *Sicherheits-Bedenken*. So schrieb er in die Lizenz einige übertriebene Einschränkungen, was die Verbreitung seiner Software angeht. Inzwischen erklärte er all seine Werke als gemeinfrei, was in den USA möglich ist (Public Domain). Seitdem ist `Qmail` freie Software. [Wik08e]

Die Autoren von `GPU` (Global Processing Unit) haben *moralische Bedenken*. Ihre Software untersteht der GPL, jedoch mit dem ersten Asimovschen Gesetz als Zusatz. Somit ist die Software nicht zu jedem Zweck einsetzbar, und daher keine Freie Software. [GPU06]

`Xv` ist ein Programm, das sich *wie Freie Software anfühlt*: Man kann den kompletten Quellcode als `tgz`-Archiv von der Homepage herunterladen, entpacken und mit `./configure; make` compilieren. Dennoch handelt es sich um Shareware. [Bra06]

6 Fortsetzung: DC-Rap

DC-Rap ist für RedHat-Linux entwickelt, daher wird jedes Release als RPM-Paket ausgeliefert. Zusätzlich werden auch der Quellcode, die Dokumentation sowie alle benötigten Zusatzpakete mitsamt Quellcode mitgeliefert.

Dies geschah vor allem zum Zwecke der Kundenzufriedenheit, doch eines der Zusatzpakete musste ohnehin mitsamt Quellcode zu Verfügung gestellt werden, da es unter GNU LGPL⁷ steht. Die übrigen Zusatzpakete sind ebenfalls freie Software, mit unterschiedlichen Lizenzen: Eines steht unter BSD-Lizenz (3-Klausel), eines unter MIT/X11-Lizenz und eines ist sogar gemeinfrei (Public Domain).

DC-Rap an sich ist keine Freie Software, jedoch sichert der Werkvertrag dem Auftraggeber ein ausschließliches, uneingeschränktes und sublizenzierbares Nutzungsrecht zu. Damit besitzt zumindest der Auftraggeber alle vier Anwenderfreiheiten, und es wäre ihm theoretisch möglich, DC-Rap zu einer Freien Software zu machen.

Zur Entwicklung von DC-Rap wurde ausschließlich freie Software verwendet. Als IDE kamen Eclipse und Vim zum Einsatz, als Compiler der GCC, als Debugger gdb und KDbg, als Profiler Valgrind und KCacheGrind, als Versionskontrolle Mercurial, und so weiter.

DC-Rap hat somit große Verbindungen zu Freier Software, obwohl es proprietär ist. Es wurde mit Freier Software entwickelt, benutzt Freie Software, läuft auf einem freien Betriebssystem und der Auftraggeber hat alle Anwenderfreiheiten.

⁷GNU Lesser General Public License, eine Variante der GNU GPL

7 Schlussfolgerungen

Das Recht und speziell Verträge spielen auch in der Software-Entwicklung eine wichtige Rolle. Es regelt das Verhältnis zwischen dem Auftraggeber und dem Auftragnehmer, und auch das Verhältnis zwischen dem Auftragnehmer und dessen Entwicklern.

Wie das Beispiel DC-Rap zeigt, kann ein Vertrag keine termingerechte Fertigstellung der Software garantieren. Obwohl das Pflichtenheft mit dem Wissen eines entwickelten Prototyps im Hinterkopf beschrieben wurde, konnte es keine nachträglichen Änderungswünsche des Kunden verhindern.

Jedoch hätte der Vertrag für eine Absicherung gesorgt, wenn das Projekt schief gelaufen wäre. Doch solange der Auftraggeber und Auftragnehmer miteinander zufrieden sind, hat der Vertrag nur eine geringe Bedeutung.

Ohne das gegenseitige Vertrauen hätte es jedoch ganz anders kommen können. So hätte beispielsweise der Auftraggeber knallhart auf die Termineinhaltung bestehen können und die bis dahin erstellte unfertige Version als mangelhaft ablehnen können. So hätte der Auftragnehmer im Preis herunter gehen müssen, und hätte keinen finanziellen Spielraum mehr gehabt, um auf wichtige Änderungswünsche einzugehen. Es wäre eine Software „stur nach Pflichtenheft“ entstanden, die letztendlich für den Auftraggeber nur einen sehr kleinen Wert gehabt hätte. Insbesondere hätte die Software einen Großteil der vom DICOM-Standard abweichenden Bilddaten nicht verarbeiten können, da die hierzu nötigen Details im Pflichtenheft nicht spezifiziert wurden. Da die Abweichungen unsystematisch und nicht vorhersehbar waren, konnten sie auch gar nicht spezifiziert werden.

Natürlich hätte das gesamte Entwicklungsverfahren auch flexibler im Vertrag festgehalten werden können, sodass die Entwicklung auch gegenseitiges Vertrauen zum Ziel geführt hätte. Alle Eventualitäten schriftlich zu erfassen wäre jedoch ein immenser Aufwand gewesen, was zu bedeutenden Mehrkosten auf beiden Seiten geführt hätte. In der Praxis lohnt sich dieser Aufwand nur für sehr große Projekte in Millioenhöhe wie zum Beispiel Regierungsaufträge. Bei kleineren Projekten wie DC-Rap wäre dieser Aufwand unverhältnismäßig hoch im Vergleich zum einfachen Aufsetzen eines Standard-Vertrages.

Im kleinen Rahmen haben Verträge somit lediglich eine unterstützende Funktion, und gegenseitiges Vertrauen ist essenziell für den Projekterfolg. Jedoch muss Vertrauen auch aufgebaut werden. Im DC-Rap-Projekt geschah dies durch eine erfolgreiche Zusammenarbeit zwischen Bayer Schering und Rent a PACS in vorangegangenen Aufträgen.

Bei neuen Projektpartnern kann eine Absicherung im Vorfeld verhindern, dass dieser notwendige Vertrauensvorschuss missbraucht wird.

Literatur

- [BGB08] BGB: *Bürgerliches Gesetzbuch*. <http://bundesrecht.juris.de/bgb/>, 2008.
- [Bra06] BRADLEY, JOHN: *Xv – Homepage*. <http://www.trilon.com/xv/xv.html>, 2006.
- [ESt08] EStG: *Einkommensteuergesetz*. <http://bundesrecht.juris.de/estg/>, 2008.
- [Fre07] FREE SOFTWARE FOUNDATION: *The Free Software Definition*. <http://www.fsf.org/licensing/essays/free-sw.html>, 2007.
- [GPU06] GPU PROJECT: *Discussion about modified GPL*. http://sourceforge.net/forum/forum.php?forum_id=601861, 2006.
- [Gra07] GRABSCH, VOLKER: *Freie Software*. http://www.bephpug.de/fohlen/freie_software.html, 2007.
- [Gra08] GRABSCH, VOLKER: *DC-Rap – Technischer Import von medizinischen Bilddaten*. <http://www.prof.v.de/dc-rap/>, 2008.
- [lex07a] LEXEXAKT.DE: *Gefahrübergang*. <http://www.lexexakt.de/glossar/gefahuebergang.php>, 2007.
- [lex07b] LEXEXAKT.DE: *Werk*. <http://www.lexexakt.de/glossar/werk.php>, 2007.
- [Ott06] OTTO, DIRK: *Recht für Softwareentwickler*. Galileo Press, Bonn, 2006.
- [Sch06] SCHULZE, MARTIN: *Freie Kommunikation gesichert – Debian sorgt für freie Lizenz*. <http://www.infodrom.org/~joey/Writing/Linux-Magazin/ircII.html>, 2006.
- [SGB08] SGB 5: *Sozialgesetzbuch, Fünftes Buch*. http://bundesrecht.juris.de/sgb_5/, 2008.
- [Urh08] URHG: *Urheberrecht*. <http://bundesrecht.juris.de/urhg/>, 2008.
- [VDI06] VDI/VDE INNOVATION + TECHNIK GMBH: *Vertragsrecht – Softwareentwicklung*. <http://www.gruenderleitfaden.de/recht/vertragsrecht/softwareentwicklung.html>, 2006.

- [Wik08a] WIKIPEDIA: *Arbeitnehmer*. <http://de.wikipedia.org/w/index.php?title=Arbeitnehmer&oldid=46617113>, 2008.
- [Wik08b] WIKIPEDIA: *Freier Mitarbeiter*. http://de.wikipedia.org/w/index.php?title=Freier_Mitarbeiter&stableid=46237846, 2008.
- [Wik08c] WIKIPEDIA: *Gemeinfreiheit*. <http://de.wikipedia.org/w/index.php?title=Gemeinfreiheit&oldid=46855346>, 2008.
- [Wik08d] WIKIPEDIA: *Kaufvertrag*. <http://de.wikipedia.org/w/index.php?title=Kaufvertrag&stableid=46825056>, 2008.
- [Wik08e] WIKIPEDIA: *Qmail – Copyright status*. http://en.wikipedia.org/w/index.php?title=Qmail&oldid=209643411#Copyright_status, 2008.
- [Wik08f] WIKIPEDIA: *Recht*. <http://de.wikipedia.org/w/index.php?title=Recht&stableid=46814910>, 2008.
- [Wik08g] WIKIPEDIA: *Urheberrecht*. <http://de.wikipedia.org/w/index.php?title=Urheberrecht&oldid=46747221>, 2008.
- [W607] WÖRTERBUCHNETZ: *Recht*. <http://www.woerterbuchnetz.de/woerterbuecher/dwb/wbgui?lemid=GR02054>, 2007.