

Wissen

Freeware, Shareware und Public Domain

von Robert Gehring

Begriffe wie Freeware, Shareware oder GNU geistern in aller Munde. Was es damit auf sich hat und wie diese Begriffe entstanden sind, erläutert uns Robert in einer kleinen Artikelreihe.

In letzter Zeit tauchen an verschiedener Stelle häufiger Fragen im Zusammenhang mit 'Shareware', 'Freeware' und 'Public Domain'-Software auf. Die Hauptursache für die darin deutlich werdenden Unsicherheiten ist sicher in dem Mangel an Informationen über die genannte Software zu sehen. Damit sind nicht Unklarheiten über die aktuelle Programmversion gemeint, sondern Fragen der Art „Was ist eigentliche Shareware?“, „Das Programm 'xyz'

ist Freeware, darf also nicht in der Firma eingesetzt werden, oder?“ und „Wer haftet eigentlich für den Schaden, den das Programm Supercash' in der Vereinskasse angerichtet hat, weil es nur mit ganzen Zahlen rechnen kann?“. Ein regelmäßiger Blick in die Newsgruppe *'de.soc.recht'* verschafft einen Überblick über das Spektrum der Problematik. Verwirrung stiften dabei insbesondere auch diverse Fachzeitschriften mit ihrem inflationären Gebrauch

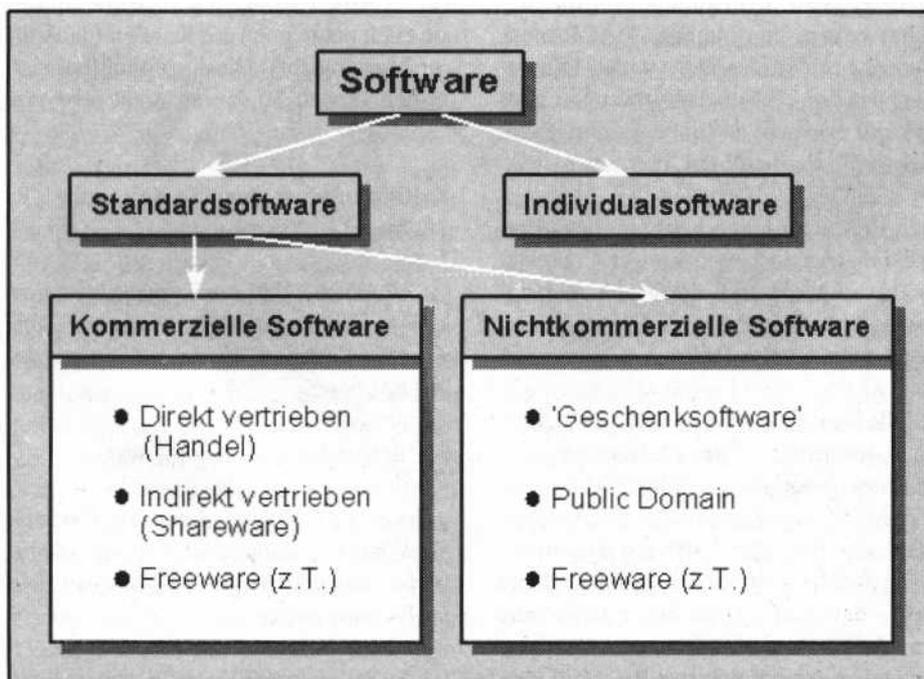


Abbildung 1: Einteilung der Software

der oben genannten Begriffe. 'Rechtsratgeber' in gedruckten und Online-Zeitschriften äußern sich zwar ebenfalls von Zeit zu Zeit zum Thema, landen dabei mit ihren Aussagen aber nur selten einen Treffer.

Jeder 'Linuxer' ist nun in besonderem Maße mit den Eigenarten von 'Freeware', 'Shareware' und 'Public Domain'-Software (im weiteren als PD abgekürzt) in Berührung gekommen: Er hat ein 'freies Betriebssystem' auf seinem Rechner installiert. Daß es sich um ein solches handelt, kann er den mitinstallierten Lizenzen, zum Beispiel der 'General Public License' (GPL), entnehmen. Viele finden auf den CD's ihrer Linux-Distributionen dann noch diverse Demos kommerzieller Programme und oft auch 'Shareware'. Die Unterschiede dieser Varianten von Software und die resultierenden Konsequenzen für Gebrauch und Weitergabe, werden wohl den wenigsten klar sein - Grund genug, in einer kleinen Artikelserie gerade im LINUX-Magazin näher darauf einzugehen.

Überblick über die Themen

Der Mangel an Informationen und die Schwierigkeit, an die wenigen vorhandenen Informationen heranzukommen, sind die Hauptursache für das Wirrwarr an Vorstellungen zu den genannten Software-Typen. An erster Stelle steht deshalb ein kurzer historischer Abriss zur Entstehung dieser Software. Daran schließt sich ein Abschnitt an, in dem übersichtsartig dargestellt wird, welche rechtlichen Dimensionen von Software im allgemeinen berührt werden. Stichpunkte hierzu sind Urheberrecht, Haftungsrecht und bürgerliches Recht. Damit soll ein Rahmen erarbeitet werden, innerhalb dessen dann einzeln und ausführlich auf 'Freeware', 'Shareware' und 'PD' eingegangen wird. Da es sich um einen Themenbereich handelt, der den meisten Informatikern nicht sehr vertraut ist, werden zu den wesentlichen Punkten Grafiken und Tabellen zur Verdeutlichung der Zusammenhänge verwendet.

Freeware

Eine kurze Geschichte der Software

In der Abbildung 1 wird eine Einordnung von Software angegeben, die aus heutiger Sicht sinnvoll ist, noch vor einigen Jahren jedoch undenkbar war und wahrscheinlich bloß Gelächter hervorgerufen hätte. 'Nichtkommerzielle' Software ist wesentlich ein Kind des Internets und das ist ja bekanntlich erst in den 70'er Jahren entstanden. Doch wir müssen noch weiter in die Vergangenheit zurückgehen, wenn wir das grundsätzlich Andere an 'Freeware' etc. verstehen wollen.

In den 50'er Jahren, der Zeit des Beginns eines industriellen Computereinsatzes, gehörten Hardware und Software untrennbar zusammen und hießen 'mainframes'. Auf dem Computergehäuse stand der Name IBM und die Software kam von ... IBM. Die Preise für ein Computersystem waren horrend und wurden von IBM bestimmt. Die Software war ebenfalls teuer, wurde nur zusammen mit dem Rechner verkauft und machte den Monopolisten IBM reich. Kundendienst wurde damals als Dienst des Kunden an der Firma IBM verstanden. So durften in IBM-Computer nur IBM-Zubehörteile eingebaut werden und nur von IBM autorisierte Software verwendet werden (die von IBM kam). Im Zusammenhang mit Big Blue's Preisgestaltung fielen Worte wie 'räuberische Erpressung' u.ä. Nachdem dies zwei Jahrzehnte lang hingenommen wurde (mangels Alternative), rang sich der US-Justizminister 1972 dazu durch, ein Antitrust-Verfahren gegen die IBM einzuleiten. Das Ergebnis ist bekannt: die Firma wurde verurteilt, Hardware- und Softwaregeschäft zu trennen. Dies könnte man als die offizielle Geburtsstunde der Softwareindustrie ansehen.

Ebenfalls zu jener Zeit wurde durch die Ausbreitung von Computern an den Hochschulen und Universitäten -insbesondere in den USA- der Kreis derjenigen, die Softwareentwicklung betrieben, immer größer. Andererseits fielen die Preise für Hardware auf erträgliche Höhen, da neue Anbieter in den Markt drängten. Die Entwicklung der Technologien wurde intensiv vorangetrieben und gipfelte 1970 in der Vorstellung des ersten Mikroprozessors

der Welt durch die Firma INTEL. Das war der 4004, ein 4-Bit-Prozessor. In der Folgezeit wurde ein neuer Rechner, basierend auf Mikroprozessoren entwickelt, der passenderweise den Namen 'microcomputer' erhielt entwickelt. Darauf lief dann ein sogenanntes 'proprietäres' Betriebssystem, d.h. jeder Hardwarehersteller lieferte sein eigenes System mit.

Die Software, die man einsetzen wollte, mußte in der Regel aber noch selbst entwickelt werden. An den Universitäten wurden in den 70'er Jahren begonnen, in Projekten eigene Betriebssysteme, Programmiersprachen und Programme zu entwickeln, die z.T. auch außerhalb der Unis Verbreitung fanden (Erinnert sich noch jemand an das UCSD-Pascal-System?). Auch in verschiedenen Firmen gab es Forschungs- und Entwicklungsabteilungen, die sich mit Software beschäftigten. Der Schwerpunkt lag allerdings immer auf einem firmeninternen Einsatz der Software. (*Aus solch einem Projekt namens MULTICS ging bei der Firma AT&T UNIX hervor aber das wird ja bekannt sein, oder?*)

Etwa zeitgleich entwickelte das Militär in den USA die Idee eines Informationssystems, das auch durch einen Atomkrieg nicht zu zerstören sein würde. Der Wunsch der 'Kalten Krieger' ging mit Hilfe von Steuergeldern in Erfüllung und das Resultat bekam den Namen 'DARPA-net'. Bereits nach kurzer Zeit wurden Universitäten in den USA an das Netz angeschlossen und es wurde so (unbeabsichtigt) zur Keimzelle des heutigen Internets.

Den Privatmenschen betrafen diese Entwicklungen jedoch eher weniger. Computer in Privathand waren die große Ausnahme, dafür sorgten Preise und Lizenzpolitik der wenigen Softwarefirmen.

All die hier aufgezählten Fakten scheinen nicht unbedingt in einem Zusammenhang zu stehen, der für die Entwicklung des 'Freeware'-Konzeptes, der 'Shareware'-Idee oder der 'PD'-Software wesentlich wäre. Sie alle waren aber ausschlaggebend dafür, daß Anfang der 80'er Jahre eine Revolution in der Softwareentwicklung begann, in deren Folge zum Beispiel Linux

entwickelt wurde. Zu Beginn der 80'er trafen aufeinander:

- erschwingliche Computer für 'jedermann',
- die frustrierenden Erfahrungen diverser Programmierer mit Soft- und Hardwarefirmen,
- die Möglichkeit, Informationen (also auch Programme) in sekundenschnelle elektronisch verbreiten zu können,
- der Beginn der Vermarktung von UNIX, das vorher frei war,
- der Einzug von Computern in die Alltagskultur und
- der Idealismus privater Softwareentwickler.

Revolutionäre

Zwei Namen stehen an allererster Stelle: **Richard Stallman** und **Jim Knopf**. Beide fingen etwa zur gleichen Zeit (1981) an, ihre Ideen davon, was Software sein sollte, in die Tat umzusetzen. Dabei verfolgten sie durchaus unterschiedliche Zielstellungen, die aber beide in krasssem Gegensatz zur damals etablierten Vorstellung von Software standen. Damals war Software etwas, was man ungeprüft kauft und mit dem man hinterher irgendwie klarkommen muß. Was erlaubt ist, und das ist nicht viel, bestimmt der Softwarehersteller und muß der Käufer hinnehmen. Mit dieser Vorstellung sind verschiedene Softwarefirmen groß geworden und verwirklichen sie auch heute noch am Kunden (dank ihrer Marktmacht). Diesbezüglich hat sich in den letzten 30 Jahren nicht sehr viel verändert.

Kritikpunkte waren für den einen (R. Stallman)

„die Tatsache, daß der Softwarehersteller den Anwender vom Wissen um die Software fernhält und über den Umgang damit bestimmt..“

Für den anderen (J. Knopf) war es

„das große Defizit, daß man die Katze im Sack kaufen mußte und trotz des damit verbundenen Risikos noch relativ viel Geld hinblättern mußte..“

Beide, Stallman und Knopf, hatten im Sinn, dem Anwender zu größerer Unabhängigkeit

von der Softwareindustrie zu verhelfen. Mittel zum Zweck war dabei die Software. Das war die wesentliche Gemeinsamkeit zwischen den beiden, die voneinander nichts wußten. Der wesentliche Unterschied zwischen beiden war auf der einen Seite die Idee der Informationsfreiheit (Stallman) und auf der anderen Seite die der Entscheidungsfreiheit beim Softwareerwerb (Knopf). Dieser Unterschied hat in der Konsequenz dann zu verschiedenen Konzepten für den Umgang mit Software geführt: 'Freeware' und 'Shareware'.

Der dritte im Bunde der 'Revolutionäre' hat keinen Namen, oder aber viele Namen. Die 'Public Domain'-Software hat andere Ursprünge und keinen einzelnen 'Fürsprecher'. PD sind entweder Programme, deren Copyright verfallen ist, oder deren Schöpfer/Copyright-Inhaber sie in die Public Domain' gegeben haben. Was das im einzelnen bedeutet, wird im entsprechenden Abschnitt genauer untersucht. Festhalten kann man an dieser Stelle, daß es sich um nichtkommerzielle Software handelt, die insbesondere durch das Internet sehr schnell Verbreitung gefunden hat.

Soweit zur Evolution der Softwareentwicklung. Jetzt werden wir uns für eine Weile einer anderen Seite von Software zuwenden und das ist die rechtliche.

Software und Recht

Wir begeben uns auf ein Terrain, das für den Informatiker meist eine 'terra incognita' darstellt. So wird es hilfreich sein, an erster Stelle einige prinzipielle Ausführungen zu machen, um eine Art Landkarte zu entwerfen. Diese soll zur Orientierung dienen, wenn später detaillierte Untersuchungen folgen.

Es gibt eine Reihe von Gesetzeswerken, die Software und Softwareentwickler mittelbar oder unmittelbar betreffen. Dies ist zum einen der geschriebene Teil des Softwarerechtes (siehe Abbildung 2). Dazu gibt es noch einen zweiten Teil, der sich aus der Rechtsprechung zu Software herleitet. Der dritte Teil ist das Gewohnheitsrecht, das sich aus der sogenannten 'Verkehrsanschauung' oder auch dem üblichen 'Geschäftsgebahren' im Umgang mit Software ergibt. Alle drei Teile in ih-

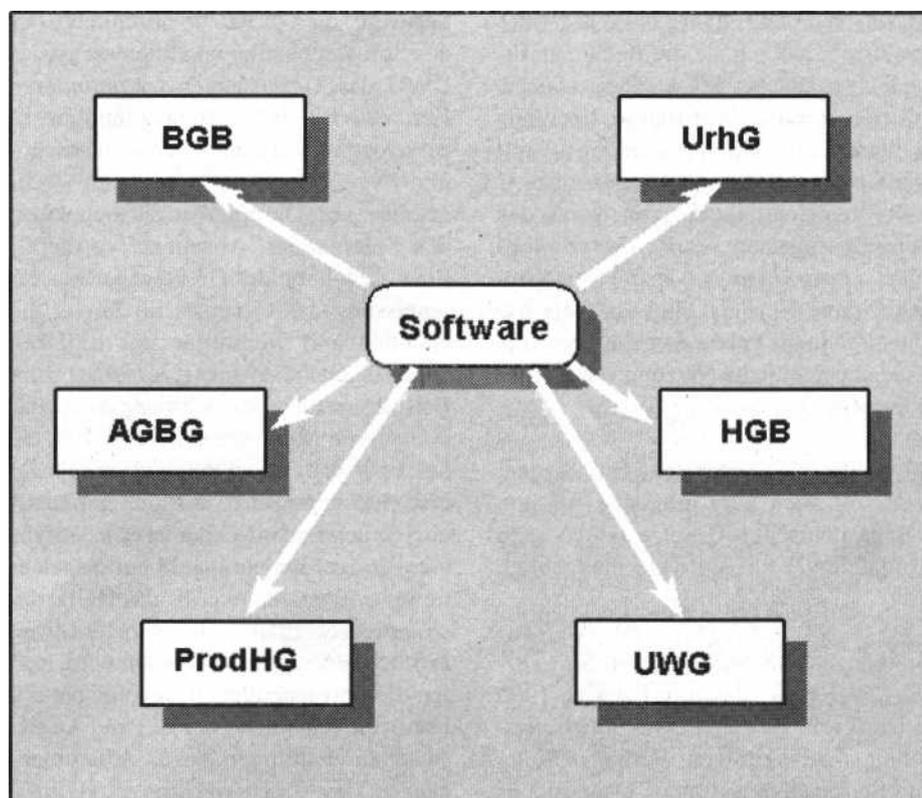


Abbildung 2: Gesetze zum Software-Recht

rer Gesamtheit bilden das Softwarerecht, das nach deutscher Rechtsauffassung vollständig ist (Abgeschlossenheitsprinzip).

Die Bedeutung dieses Prinzips der Abgeschlossenheit ist darin zu sehen, daß prinzipiell davon auszugehen ist, daß alle Rechtsfragen mit diesem Recht zu beantworten sind. Ein anderes Recht, insbesondere ausländisches Recht, kann nicht herangezogen werden (im Prinzip), lediglich internationales Recht spielt eine Rolle. Für unsere konkrete Fragestellung nach der Stellung von Freeware, Shareware und PD im deutschen Recht bedeutet das, daß das Recht des Ursprungslandes dieser Softwaretypen nicht zur Bewertung beitragen kann. Ein anderer Grund dafür ist der unterschiedliche Ursprung deutschen und amerikanischen Rechtes. Ist das deutsche Recht ein Produkt des 'Naturrechtsgedankens', folgt das amerikanische Recht dem 'common law', das sich in England entwickelt hat. Die für unser Problem wesentlichen Unterschiede liegen dabei in den Konzepten vom Copyright ('common law') und Urheberrecht (Naturrecht), sowie auch in haftungsrechtlichen Fragen.

- Im 'common law' gibt es den Leitge-

danken, daß bei Leistung ohne Gegenleistung (z.B. bei verschenkter Software) der Leistungserbringer nicht für Schäden, die aus seiner Leistung (z.B. Software) resultieren, haftbar gemacht werden kann. Solch einen Grundsatz kennt das deutsche Recht nicht!

- Der Unterschied zwischen Copyright und Urheberrecht ist im wesentlichen der, daß das Copyright im Sinne eines Besitzrechtes gilt, d.h. es kann übertragen werden (verschenkt, verkauft, etc.), während das (deutsche) Urheberrecht quasi ein Persönlichkeitsrecht ist, das an die Person des Urhebers gebunden ist (mit der Ausnahme der Vererbung).

Letzterem Umstand war es auch geschuldet, daß die USA dem international maßgeblichen Vertragswerk zu Urheberrechtsfragen, der 'Erweiterten Berner Übereinkunft', erst am 1. März 1989 beigetreten sind. Seit diesem Tag genießen in- und ausländische Urheber in den USA den Schutz, den sie in fast allen anderen Staaten (Ausnahme u.a. China) z.T. bereits seit 1886 in Anspruch nehmen konnten. Bis zu diesem Zeitpunkt mußten Ausländer und Amerikaner in den USA ihren Rechtsan-

Freeware

spruch beim Copyright-Office registrieren lassen, wollten sie die Rechte an ihren Werken (Bücher, Musik, Theaterstücke, Software, etc.) nicht verlieren. Ebenfalls mußten sie alle Kopien ihrer Werke mit dem Copyright-Hinweis (eingekreistes C - © , Veröffentlichungsjahr, Name des Copyright-Inhabers - z.B.: *Copyright (D) 1996, Linux-Magazin Verlag*) versehen. Fehlte diese Hinweis, ging das Werk automatisch in die public domain' über, d.h. seine unentgeltliche Nutzung war frei für jedermann.

So viel zu Konzepten und Grundlagen. Jetzt soll noch kurz umrissen werden, welche deutschen Gesetze (siehe dazu Abbildung 2) für Software maßgeblich sind.

Grundlegend ist das BGB, das '*Bürgerliche Gesetzbuch*', das zum 1. Januar 1900 in Kraft trat. Es enthält wesentliche Regelungen zu Verträgen, Vertragshaftung und Entschädigungsfragen. Dann gibt es das AGBG, dessen vollständige Bezeichnung '*Gesetz zur Regelung des Rechts der Allgemeinen Geschäftsbedingungen*' lautet. Darin wird u.a. das sogenannte Kleingedruckte' in Kaufverträgen behandelt. Es gilt für alle Arten von Kaufverträgen über

Software, die ein Kunde abschließt. Für den Softwarehändler von Interesse ist das UWG, das '*Gesetz gegen den unlauteren Wettbewerb*'. (In welchem Sinne dieses ausgelegt werden kann, führen mit unschöner Regelmäßigkeit sogenannte 'Abmahner' vor. Dafür haben sie sich schon den Spitznamen 'Absahner' verdient.) Eigentlich hatte der Gesetzgeber bei der Abfassung des Gesetzes im Sinne, die Händler und Hersteller vor unfairen Geschäftspraktiken ihrer Konkurrenz -zum Beispiel irreführender Werbung oder Mißbrauch von Markennamen- zu schützen. Leider ist das Gesetz durch seine Mängel bloß noch eine Karrikatur dieser Intention. Für den Endanwender ist es jedenfalls von geringer Bedeutung. Ebenfalls wichtig für Softwarehändler ist das HGB, das '*Handelsgesetzbuch*'. Insofern Software in Stückzahlen ge- und verkauft wird, treffen die dort enthaltenen Regelungen auf Kaufleute (Händler) zu. HGB und AGBG stehen in vielfältigen, engen Zusammenhängen. Das '*Urheberrechtsgesetz*' (UrhG) legt fest, welche Rechte die Entwickler von Software haben und in welchem Rahmen Software verwendet werden darf. Die Bedeutung dieses Gesetzes liegt in der wirtschaftlichen Absicherung der Schöpfer von geistigen Werken, wozu Software

ebenfalls gezählt wird. Zuletzt ist dann noch das '*Produkthaftungsgesetz*' (ProdHG) zu erwähnen, das die Haftung für Schäden klärt, die irgendwelche Produkte (Produkte im Sinne dieses Gesetzes; landwirtschaftliche Erzeugnisse zählen z.B. nicht dazu!) beim Verbraucher anrichten. Es soll eine Hemmschwelle für Produzenten sein, um zu verhindern, daß diese ihre Produkte leichtfertig auf den Markt bringen.

Genug der rechtlichen Theorie. Jeder sollte jetzt eine ungefähre Vorstellung davon haben, an welchen Punkten der Einsatz von Software zu rechtlichen Fragen führen kann. Was davon konkret auf Freeware, Shareware und PD zutrifft, wird im nächsten Artikel dieser Folge detaillierter dargestellt werden.

Der Autor

Robert Gehring ist Student der Informatik und Philosophie an der TU Berlin. Seit der Linux-Version 0.9.12 ist er mit dabei, entwickelte diverse Software und setzt dieses Betriebssystem auch ausgiebig für seine Studien/Diplomarbeit ein. Zu erreichen ist er unter rag @zblmath. FIZ-Karlsruhe.DE.