

Softwarelizenzen und Open Source

Michael Woborschil

Juristisches IT-Projektmanagement WS 2015/16

Dozent: Dr. Frank Sarre

Ludwig Maximilians Universität München

Zusammenfassung:

Bei der Erstellung von Software nutzt man in den allermeisten Fällen bereits bestehende Software um den Entwicklungsprozess zu vereinfachen. Oft handelt es sich hierbei um sogenannte Open Source Software. Diese kann unter vielen verschiedenen Softwarelizenzen stehen, die den Umgang und die Verbreitung der mit ihr erstellten Software regelt. Um aber als Open Source durchzugehen, müssen all diese Lizenzen bestimmte Kriterien der Open Source Initiative erfüllen. Wichtige Beispiele solcher Softwarelizenzen sind die Mozilla Public License, die Eclipse Public License und die GNU General Public License.

Inhaltsverzeichnis

1. Verwendung von bestehender Software in eigenen Projekten.....	2
2. Lizenzen allgemein.....	2
3. Open Source.....	3
4. Copyleft.....	5
5. Beispiele für Softwarelizenzen.....	6
5.1. WTFPL.....	6
5.2. MPL: Mozilla Public License.....	6
5.3. EPL: Eclipse Public License.....	7
5.4. GNU General Public License.....	7
6. Kompatibilität von Softwarelizenzen.....	8
7. Quellen.....	9

1. Verwendung von bestehender Software in eigenen Projekten

Bei der Entwicklung von Software ist die Verwendung bestehender Software unabdingbar. Ob bestimmte Entwicklungsumgebungen bei der Programmierung verwendet werden, oder ob man auf bereits bestehenden Programmen aufbaut und deren Quellcode modifiziert um eine eigene Software zu erhalten, in beiden Fällen gilt es die Lizenz, unter der die verwendete Software herausgegeben wurde zu kennen, um gewisse Fragen beantworten zu können. Beispielsweise unter welchen Umständen man seine damit erstellten Programme verbreiten darf und wie weit man die Rechte der Nutzer bezüglich weiterer Verwendung einschränken darf, so dass das eigens erstellte Werk strengeren Vereinbarungen unterliegt wie das ursprünglich zur Entwicklung verwendete.

2. Lizenzen allgemein

Um ein Verständnis für Softwarelizenzen zu bekommen, sollte man sich über den zugrunde liegenden Begriff der Lizenz im Klaren sein. Hierfür finden sich einige Definitionen, die sich stark in ihrer Ausführlichkeit unterscheiden.

Als knappe Definition bietet sich die des Duden an:¹

„[gegen eine Gebühr erteilte] rechtskräftige Genehmigung (z. B. zur Ausübung eines Gewerbes, zur Nutzung eines Patents, zur Übersetzung oder Übernahme eines Werks)“

Eine weitaus umfassendere Definition ist im Gabler Wirtschaftslexikon zu finden:²

„Begriff: die vom Inhaber eines gewerblichen Schutzrechts oder urheberrechtlichen Verwertungsrechts einem Dritten eingeräumte Befugnis, die dem Rechtsinhaber zustehenden Verwertungsrechte auszuüben (Nutzungsrecht). Lizenzen i.e.S. sind Nutzungsrechte an den technischen Schutzrechten (Patent, Gebrauchsmuster, Sortenschutzrecht), ferner an Halbleiterrechten (Halbleiterschutzrecht) und Marken, in der Sache aber auch die Dritten eingeräumten Nutzungsrechte an Geschmacksmustern, Schriftzeichen und urheberrechtlichen Verwertungsrechten. Im Einzelnen: Berechtigung zur Nutzung von

(1) Erfindungen oder von Schutzrechten für Erfindungen (Patente),

(2) Gebrauchsmustern oder deren Anmeldungen,

(3) Marken, Copyrights,

(4) technischem Know-how (Technologie-Transfer),

(5) kaufmännischem, v.a. Marketing- und Management-Know-how; alle mit der vertraglich fixierten Erlaubnis zur Nutzung von Urheberrechten und/oder mit Know-how-Überlassungsabkommen verbundenen (v.a. absatzmarktorientierten) Planungen und Handlungen des Lizenzgebers.“

Im hier anzutreffenden Kontext der Softwarelizenzen wird es sich also um die Einräumung des Entwicklers handeln, seine Software von Dritten nutzen zu lassen und in welchem Umfang diese verändert und anschließend weitergegeben werden darf.

3. Open Source

Hinter dem Begriff Open Source steht die Open Source Initiative (OSI). Sie wurde im Februar 1998 von Eric Raymond und Bruce Perens gegründet.

Die Bezeichnung „open source“ wurde am 03. Februar 1998 bei einer Konferenz in Palo Alto in Kalifornien kurz nach Veröffentlichung des Netscape Quellcodes kreiert. Sie sollte neue Ansichten auf die freie Verwendung von Software bieten und sich zugleich vom Begriff der „free software“ abgrenzen.

Eine der ersten Aufgaben die sich die OSI machte, war die Open Source Definition (OSD) festzulegen, um auf Basis dieser eine Liste der von der OSI anerkannten Softwarelizenzen zu erstellen.

Die ursprüngliche Open Software Definition wurde von den Debian Free Software Guidelines abgeleitet, indem diese überarbeitet wurden und sich spezifisch auf Debian beziehende Stellen gestrichen wurden. Dies geschah schon während der Gründungszeit der OSI.

Im Oktober 1999 wurde die erste Liste der von der OSI anerkannten Lizenzen veröffentlicht. Seither wurde die Liste viele male aktualisiert und auf sie beziehen sich immer wieder andere Organisations, Standardisierungsgremien und sogar Regierungen.

2004 wurde die Definition von Open Source um die zehnte Klausel erweitert. Sonst wurden bisher immer nur kleine Änderungen an einzelnen Formulierungen vorgenommen. Auch im Jahr 2004 wurde auf Grund der steigenden Anzahl an Open Source Lizenzen von der OSI eine Kampagne ins Leben gerufen, die den Anstieg dieser begrenzen sollte.³

Definition von Open Source:⁴

Open Source heißt nicht nur Offenlegung des Quellcodes, sondern die Bedingungen für die Weiterverbreitung einer Open Source Software müssen auch die folgenden Kriterien erfüllen:

1. Kostenlose Weiterverbreitung

Die Lizenz darf niemanden darin einschränken, die Software zu verkaufen oder Softwaresammlungen aus mehreren verschiedenen Quellen zu verbreiten. Sie sollte zudem keine Lizenzgebühren oder andere Gebühren für einen solchen Verkauf vorsehen.

2. Quellcode

Ein Programm muss seinen Quellcode beinhalten, und es muss die Weitergabe dessen sowohl als auch in kompilierter Form erlauben. Wird ein Produkt ohne Quellcode ausgeliefert, muss es einen wohlbekanntem Weg geben, über den dieser für nicht mehr als einen angemessenen Preis für dessen Vervielfältigung bezogen werden kann. Vorzugsweise ohne Kosten über das Internet. Der Quellcode muss die bevorzugte Art sein, über die ein Programmierer das Programm modifiziert. Absichtlich verwirrender Code ist genau so verboten, wie zwischenliegende Formen wie die Ausgabe eines Präprozessors oder Übersetzers.

3. Abgeleitete Arbeiten

Die Lizenz muss Änderungen und abgeleitete Arbeiten erlauben, und muss es erlauben, dass diese unter den selben Bedingungen wie die ursprüngliche Software verbreitet werden darf.

4. Integrität des Quellcodes des Autoren

Die Lizenz darf die Verbreitung von modifiziertem Sourcecode nur dann einschränken, wenn die Lizenz das Verteilen von Patches mit Quellcode erlaubt, mit dem Ziel das Programm zur Laufzeit zu modifizieren. Die Lizenz muss ausdrücklich die Weitergabe von Software erlauben, die mit modifiziertem Quellcode erstellt wurde. Die Lizenz kann dann fordern, dass diese abgewandelte Arbeit dann einen anderen Namen oder eine andere Versionsnummer trägt als das ursprüngliche Programm.

5. Keine Diskrimination gegen Personen oder Gruppen

Die Lizenz weder eine einzelne Person noch eine Gruppe an Personen diskriminieren.

6. Keine Diskrimination bestimmter Betätigungsfelder

Die Lizenz, darf niemanden in einem bestimmten Betätigungsfeld in der Nutzung der Software einschränken. Beispielsweise darf sie die Verwendung des Programms in einer bestimmten Branche oder zur Verwendung in der Genforschung nicht einschränken.

7. Verbreitung der Lizenz

Die Rechte, die mit einem Programm gewährt werden, müssen auch jedem gewährt werden, an den dieses weitergegeben wird, ohne die Umsetzung einer zusätzlichen Lizenz der Beteiligten nötig zu machen.

8. Die Lizenz darf nicht auf ein Produkt spezifiziert sein

Die Rechte, die mit einem Programm gewährt werden, dürfen nicht davon abhängen, ob es Teil einer bestimmten Software Distribution ist. Wird das Programm aus dieser Distribution, nach den Bestimmungen der angewandten Softwarelizenz, einzeln verwendet oder weitergegeben, sollen alle Parteien, an die das Programm weiterverteilt wird, die gleichen Rechte haben, die auch in Verbindung mit der ursprünglichen Software Distribution gewährt wurden.

9. Die Lizenz darf andere Software nicht einschränken

Die Lizenz darf keine Einschränkungen von anderer Software fordern, die gemeinsam mit der lizenzierten Software verteilt wird. Zum Beispiel darf die Lizenz nicht besagen, dass alle Software, die auf dem gleichen Medium ausgeliefert wird auch Open Source Software sein muss.

10. Die Lizenz muss Technologie-Neutral bleiben

Keine Klausel der Lizenz darf Aussagen über irgendwelchen einzelnen Technologien oder Interfacedesign machen.

4. Copyleft ⁵

Die im Anschluss an diesen Abschnitt vorgestellten Softwarelizenzen unterscheiden sich in mehreren Punkten. Einer davon ist, inwieweit die Methode des Copyleft zum Einsatz kommt.

Das Konzept des Copyleft besteht darin, ein Werk (in diesem Falle Software) frei zu machen und dafür zu sorgen, dass auch alle davon abgewandelte Software frei ist.

Angenommen, man stelle eines seiner Programme in die Gemeinfreiheit, um möglichst vielen Leuten die Chance zu geben davon zu profitieren, indem sie es verwenden und modifizieren können. Dadurch kann es jedoch passieren, dass Programmierer die als freie Software zur Verfügung gestellte Version abändern und jene neue dann nicht mehr frei zur Verfügung stellen. Letzteres würde dann dazu führen, dass Nutzer der neuen und abgeänderten Version gewisse Rechte einbüßen würden, die der ursprüngliche Autor des Produkts eigentlich für alle Nutzer beabsichtigt hatte.

Um dies zu verhindern, kann man sich eines Copylefts behelfen. Copyleft erlaubt es einem Nutzer ausdrücklich das Werk zu verbreiten, egal ob in der ursprünglichen oder in abgewandelter Form, jedoch nur dann wenn er die vom Urheber festgesetzten Freiheiten weitergibt.

Um Copyleft anwenden zu können, muss man zuerst sein Urheberrecht in Anspruch nehmen. Über eine an das Programm gebundene Softwarelizenz kann nun oben genannter Anspruch rechtlich an das Programm gebunden werden, nämlich dass Modifizierung und Weitergabe des Resultats erlaubt sind, solange die Freiheiten auch weiterhin gestattet bleiben.

Copyleft steht also in engem Zusammenhang mit Copyright. Während Copyright Nutzer in der Verwendung einzelner Werke einschränkt, macht sich Copyleft das Prinzip des Urheberrechts zu Nutze, um die Freiheit der Werke zu erhalten.

5. Beispiele für Softwarelizenzen

Im Nachfolgenden werden einige Softwarelizenzen genauer betrachtet, die auf Grund ihrer Einfachheit oder ihrer großen Verbreitung aus der großen Zahl verschiedener Softwarelizenzen herausstechen.

5.1. WTFPL

Eine Softwarelizenz, die durch ihre Kürze besonders auffällt ist die WTFPL (Do What The Fuck You Want Public License). Die aktuelle Version 2 wurde von Sam Hocevar im Jahr 2004 verfasst und ihr kompletter Lizenztext lautet folgendermaßen:⁶

```
DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004
```

```
Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>
```

```
Everyone is permitted to copy and distribute verbatim or modified
copies of this license document, and changing it is allowed as long
as the name is changed.
```

```
DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
```

```
0. You just DO WHAT THE FUCK YOU WANT TO.
```

Die WTFPL wird in dieser Version von der Free Software Foundation als GPL (siehe 4.4.) kompatibel anerkannt. Da sie nur aus einer Regel besteht, und diese nicht besonders umfassend ist, wird die WTFPL gerne kritisiert, da es rechtlich nicht geklärt ist, ob diese im Klagefall als rechtbindlich standhält und sie kein Copyleft beinhaltet.⁷

5.2. MPL: Mozilla Public License^{8 9 10}

Die Mozilla Public License (MPL) ist eine von der Mozilla Foundation veröffentlichte Softwarelizenz. Sie wurde 1998 von Mitchell Baker entwickelt.

Die aktuelle Version 2.0 wurde 2012 veröffentlicht, wurde von der OSI als Open Source Lizenz anerkannt und wird hauptsächlich zur Lizenzierung des Mozilla Firefox und dazugehöriger Software benutzt.

Sie beinhaltet nur ein schwaches Copyleft, was bedeutet, dass veränderte Dateien zwar weiterhin frei bleiben müssen, neu hinzugefügte jedoch beliebig lizenziert werden dürfen.

5.3. EPL: Eclipse Public License ^{11 12}

Die Eclipse Public License basiert auf der Common Public License (CPL), die hauptsächlich für die Lizenzierung der Entwicklungsumgebung Eclipse und zugehörigen Plugins zum Einsatz kommt. Im April 2009 beschlossen die Eclipse Foundation und IBM, dass die EPL die CPL ablöst, um die hohe Anzahl von Open Source Lizenzen zu verringern.

Die Eclipse Public License wird von der Open Source Initiative als Open Source Lizenz anerkannt.

Wie die MPL, so beinhaltet auch die EPL nur ein schwaches Copyleft, womit zwar modifizierte Programmteile weiterhin frei sein müssen, neue jedoch beliebig lizenziert werden können.

5.4. GNU General Public License ^{13 14 15 16}

Die GNU General Public License ist die weitverbreitetste Open Source Lizenz. Sie wurde im Januar 1989 vom Gründer des GNU-Projekts Richard Stallman geschrieben. 1991 wurde Version 2 veröffentlicht und seit dem 29. Juni 2007 ist nun Version 3 die Aktuellste.

Stallmans Ziel war aus verschiedenen damaligen Lizenzen, die auf einzelne Programme spezialisiert waren, eine Lizenz zu entwickeln, die bei jeder Software eingesetzt werden kann.

Als Grundlagen der GPL gibt es vier Freiheiten, die ein Programm als freie Software auszeichnen:¹³

- *die Freiheit, das Programm für jeden Zweck auszuführen,*
- *die Freiheit, das Programm eigenen Bedürfnissen anzupassen,*
- *die Freiheit, die Software mit Freunden und Mitmenschen auszutauschen und*
- *die Freiheit, gemachte Änderungen mit anderen auszutauschen.*

Die GNU GPL war die erste Softwarelizenz für den allgemeinen Gebrauch, die Copyleft unterstützt. Durch ihre Aktualisierung soll sichergestellt werden, dass weder technische noch rechtliche Entwicklungen das beinhaltete Copyleft aushebeln können. Generell ist der Gedanke des Copyleft bei der GNU GPL stark ausgeprägt, was bedeutet, dass im Gegensatz zu EPL und MPL selbst bei neu hinzugefügten Dateien die Regeln der GNU GPL angewendet werden sollen.

6. Kompatibilität von Softwarelizenzen ¹⁷

Zum Abschluss kann man sich noch die Frage stellen, was passiert, wenn man für ein Projekt mehrere lizenzierte Codebestandteile verwendet, die durch verschiedene Softwarelizenzen geschützt sind. Dann ist es nämlich nicht pauschal möglich eine Aussage zu machen, ob diese gemeinsame Nutzung lizenzrechtlich machbar ist. Wenn für eine Software Bestandteile aus einem Projekt X, das unter Lizenz L1 steht und Bestandteile aus einem Projekt Y, das unter Lizenz L2 steht zusammengeführt werden sollen, müssen sowohl L1 als auch L2 dies genehmigen. Ist das der Fall, spricht man von Softwarekompatibilität.

Gerade bei Verwendung von Copyleft ist eine Kompatibilität oft schwierig, da Lizenzen wie die GPL fordern, dass modifizierte Software wieder unter den Bedingungen der GPL weitergegeben werden sollen. Es gibt zwei Bedingungen, die eine Aussage zulassen, ob eine Kompatibilität solcher Open Source Lizenzen vorliegt. Zwei Lizenzen sind dann miteinander kompatibel, wenn eine der Bedingungen zutrifft:

- Enthält eine von beiden Lizenzen keine Lizenzpflichten, die nicht bereits von der Anderen vorgesehen werden, dann sind beide kompatibel und das Projekt kann unter der stärkeren Lizenz vertrieben werden.
- Enthält eine Open Source Lizenz eigene Kompatibilitäts- bzw. Öffnungsklauseln gegenüber der zweiten Copyleft-Lizenz, so sind diese miteinander kompatibel.

7. Quellen

- 1 Duden | Lizenz | Rechtschreibung, Bedeutung, Definition, Synonyme, Herkunft
<http://www.duden.de/rechtschreibung/Lizenz>
aufgerufen am 31.01.2016
- 2 Definition >> Lizenz << | Gabler Wirtschaftslexikon
<http://wirtschaftslexikon.gabler.de/Definition/lizenz.html>
aufgerufen am 31.01.2016
- 3 History of the OSI | Open Source Initiative
<https://opensource.org/history>
aufgerufen am 31.01.2016
- 4 The Open Source Definition | Open Source Initiative
<https://opensource.org/osd>
aufgerufen am 31.01.2016
- 5 Copyleft. Was ist das? - GNU Projekt – Free Software Foundation
<http://www.gnu.org/copyleft/copyleft.de.html>
aufgerufen am 31.01.2016
- 6 About the WTFPL
<http://www.wtfpl.net/about/>
aufgerufen am 31.01.2016
- 7 Verschiedene Lizenzen und Kommentare - GNU Projekt – Free Software Foundation
<http://www.gnu.org/licenses/license-list.html>
aufgerufen am 31.01.2016
- 8 Mozilla Public License, die zweite | heise open
<http://www.heise.de/open/meldung/Mozilla-Public-License-die-zweite-1403357.html>
aufgerufen am 31.01.2016
- 9 Mozilla Public License, version 2.0
<https://www.mozilla.org/en-US/MPL/2.0/>
aufgerufen am 31.01.2016
- 10 Mozilla Public License – Wikipedia
https://de.wikipedia.org/wiki/Mozilla_Public_License
aufgerufen am 31.01.2016
- 11 Eclipse Public License (EPL) Frequently Asked Questions
<https://eclipse.org/legal/eplfaq.php>
aufgerufen am 31.01.2016

- 12 Eclipse Public License – Version 1.0
<https://www.eclipse.org/legal/epl-v10.html>
aufgerufen am 31.01.2016
- 13 GNU GPLv3: Eine Kurzanleitung - GNU Projekt – Free Software Foundation
<http://www.gnu.org/licenses/quick-guide-gplv3>
aufgerufen am 31.01.2016
- 14 GNU General Public License, Version 3.0 - GNU Projekt – Free Software Foundation
<http://www.gnu.org/licenses/gpl-3.0.de.html>
aufgerufen am 31.01.2016
- 15 Frequently Asked Questions about the GNU Licences - GNU Projekt – Free Software Foundation
<http://www.gnu.org/licenses/gpl-faq>
aufgerufen am 31.01.2016
- 16 GNU General Public License
https://de.wikipedia.org/wiki/GNU_General_Public_License
aufgerufen am 31.01.2016
- 17 Was ist Lizenzkompatibilität? | ifrOSS
<http://www.ifross.org/was-lizenzkompatibilitaet>
aufgerufen am 31.01.2016