VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS ADMINISTRATION

BACHELOR THESIS

Title:

Comparing and Surveying Open Source Licenses

Author:	Daniel NIKOLL
Matriculation Number:	0250676
Field of Study:	J033 526 Bakkalaureat Wirtschaftsinformatik
Course:	1236 Vertiefungskurs VI/Bakkalaureatsarbeit – Electronic Commerce
Tutor:	ao.Univ.Prof. Dr. Rony G. Flatscher

I assure

that I have composed this bachelor thesis independently.

that I have only used quoted resources and no other unauthorised help.

that I have not submitted this thesis (neither at home nor abroad) to any judge for marking so far.

that this thesis is consistent with the marked thesis from the tutor.

Daniel
Nikoll

Digital unterschrieben von Daniel Nikoll
DN: cn=Daniel Nikoll, sn=Nikoll,
givenName=Daniel, c=AT,
serialNumber=423686726617
Grund: Ich bin der Verfasser dieses
Dokuments
Speicherort: Kirchberg/Pielach
Datum: 2006.07.06 15:53:46 +02'00'

2006-07-06

Date

Signature

Daniel Nikoll

licenses@nikoll.info

Comparing and Surveying Open Source Licenses

BACHELOR THESIS

Vienna University of Economics and Business Administration

June 2006

Abstract

The content of this Bachelor Thesis deals predominantly with Open Source Licenses. At the beginning you will get an overview about the history and the development of free and Open Source Licenses. Afterwards the most frequently used licenses are presented and described. The next point is about the connexion of the GPL with the European Law as well as an outlook on GPLv3. Finally, in the last section the economic effects like consideration of profitability, economic motives for participation, and incentives why Open Source Software should be implemented are specified.

Keywords

Open Source Licenses, Free Software, Open Source Software, Public Goods, Public Domain, Copyleft, European Law, Economy, Business Models, Motives, GNU, General Public License (GPL), Lesser General Public License (LGPL), Berkeley Software Distribution License (BSD), Apache License, Mozilla Public License (MPL), Artistic License, Common Public License (CPL), GPLv3

Table of Contents

Table of Figures

Introduction

During my study at the University of Economics and Business Administration my interest in legal aspects was highly sparked concerning the information technology. Therefore I have decided that I will devote my Bachelor Thesis to such a topic. As at the beginning of the summer term 2006 Mr. Flatscher proposed this issue I was quite pleased and started working on it.

In the first chapter the basic definitions are made like Public Domain, Copyright and License Agreements. Furthermore, both the road to the software as a product and the development of Free Software are explained and the causes for the movement and the introduction of the item Open Source are described.

To be listed as an Open Source License some criteria do have to be fulfilled which are mentioned at the beginning of the second chapter. Subsequently, a categorisation is accomplished. Afterwards, the predominantly used Open Source Licenses are explained in detail and finally the allocation of the different licenses as well as the important aspects of licenses are pointed out. The next subchapter deals with the connexion of the GPL with the European Law as well as an outlook on the current discussed version of GPL, known as GPLv3.

Finally, in the third and last section the economic effects of Open Source are specified. The possibility of earning money and studies about consideration of profitability and about the usage of Open Source in the municipality Vienna are presented. Additionally the economic motives for participating in Open Source Projects are discussed.

To make the way of reading easier only the male gender is used, though both male and female are meant.

1 Basics

1.1 Definitions

1.1.1 Freeware

The characteristic feature of Freeware is the free availability of the software the time of which is not limited. As a rule no source code is enclosed to the Freeware, partly the change is even prohibited specifically. In contrast to Open Source Software the freedom of rights is not granted, just the use according to the terms of the contract.

An example of Freeware is the so-called browser war between Netscape and Internet Explorer. Microsoft tried to give their own browser Internet Explorer away area-wide in order to squeeze the competitor out of the market [cf. O.A.06g and O.A.06h].

1.1.2 Shareware

Shareware is labelled as a form of distribution for proprietary software which is a trial version in advance and without payment. The user is allowed to copy the software in an unvaried form; however, in contrast to Freeware, only for a certain trial period free of charge. Regularly, the use of the software is disabled after the end of the trial period, unless the software is purchased and the necessary code is acquired [cf. O.A.06 et seq.].

1.1.3 Public Domain

Software is called Public Domain if there is no copyright. Basically, every intellectual work such as literary, artistic but also scientific works as well as software is subjected to the copyright. Nevertheless, in Austria and in Germany a complete renunciation of the copyright is not possible (cf. § 19 (2) UrhG). Therefore in Austria there is no Public Domain by a legal renunciation like in the USA where it is possible to renounce all kinds of rights. Nevertheless, in Austria it is possible to make a work available under such a right of use, so that it is freely changeable for everyone. It should be made aware that a Public Domain always refers to the respective national legal system. For example unprotected pictures from the US government which are Public Domain in the USA are protected by copyright in Germany [cf. O.A.06i and Müll99, 9].

1.1.4 Copyright

Every work that is not public domain underlies a copyright, or possibly some more. According to the Bernese Convention the copyright need not be declared explicitly. That means: The author of a work owns the copyright, even if the work is not provided with an explicit copyright note. In the Austrian Copyright Law this is implementated in § 10. The author of the software can put his work under several licenses. Therefore it is possible that commercial and free use is available at the same time.

In the Austrian Copyright Law there are also regulations concerning software. In § 40a it is declared that computer programmes are protected by the copyright law if those are a result of a specific intellectual creation.

In order that this law is not disabled it has to be declared who the legal owner of the work is. This can be rather difficult, in particular concerning any software project, because in a normal case a lot of people have contributed to it. In § 11 the authorship of several people is declared. It says one can talk about co-authorship, if some people who have achieved a product or work together so that each of them has contributed his own "specific intellectual creation" in order to create an inseperable whole unit [cf. Wieb04, 66 et seq.].

One speaks about inseperability if the several individual contributions are not seperately utilisable, therefore not autonomously marketable. That means that in the case of modifications of the work the consens of all co-authors is necessary [cf. Wieb04, 66 et seq.].

§ 11 (2) UrhG declares a situation in which every part is marketable as fractional authorship.

1.1.5 License Agreements

A license agreement is a contract in which the owner of a protective right or work (licenser) puts someone else (licensee) a right of use. The licenser obliges himself that a licensee can use a specified intangible, intellectual item according to the license agreement. The licenser need not give up the intangible item itself (for example the industrial property rights, intangible item) [cf. O.A.06m].

The license agreement is not declared by law. It is named as "sui generis" which is a contract for the performance of a continuing obligation. Additionally to the general legal rules special rules are applied which are appropriate for the form of contract, which focuses on the so-called "Bestandsvertrag". According to the Austrain General Civil Code this term is used for two types of contracts: rent agreements and leases [cf Wieb04, 247 et seq.].

Contracts about proprietary software also show a license-contracutal aspect. Such contracts contain different regulations either in form of general terms and conditions or individually negotiated contracts which manage the usage, distribution or modification. However, this means that pure license agreements are not used for contracts of software as this is the case with the licensing of patents or know-how [cf Wieb04, 247 et seq.].

Even if contracts of software should be explicitly designed as license agreements the legal side of intangible assets should not be overemphasised. However, the question of warranty and the contractual liability is more important wherefore the appropriation to the current agreement categories of the civil law is sustainable. Thereby the main obligation is the necessary granting of the usage rights [cf Wieb04, 247 et seq.].

All Open Source Licenses have in common that they correspond to the Open Source Definition and that there is an exclusion of liability of the programmer. All licenses are not Open Source; therefore they must not be changed!

1.2 Software as a Product

At the beginning of the computer age any software was Open Source and free. As at that time for software no own markets existed, sources were freely available. The software ran generally only on the hardware for which it was especially developed**,** and was an addition or an instruction for a concrete date processing arrangement. From the point of the computer manufacturers this also was no problem, because they received their turnovers anyway with the sales of the hardware. Therefore the software was provided by the hardware manufacturers and the users adopted the code for their own purpose. Also in magazines unlimited source code was circulating. The community was in the position to access to a growing continuance of software which was constantly extended and improved by the developers. Their payment is orientated on the achievement of the programme and not on the complete programme [cf. Horn01 and Gras04, 202 et seq.].

In the 60ies the computer market was dominated by few hardware producers like IBM, DEC and Hewlett-Packard. These companies offered complete packages, consisting of the real computer hardware, the software, the periphery devices as well as servicing and training achievements. In 1969 the situation changed when IBM began with 70 percent market share at that time, to give up the bundling of products in view of a prefaced antitrust suit. Hence, the way was laid to a decoupled independent software industry [cf. Gras04, 202 et seq.].

The programmes which had already existed at that time were considered by IBM as public domain, as those were mainly written in cooperation with the developer community. Therefore nobody can probably take up the copyright for him.

This step was vehemently criticised, as on the one hand the decoupled arrangement created a market for software, but on the other hand it was assumed that nobody was really willing to pay for such comparable products as disposable products were also available. Nevertheless, in the first half of the 70ies an independent software industry was established [cf. Gras04, 202 et seq.].

However, the decision of IBM to develop the personal computer (PC) was a watershed in the history of the software development. At that time it was decided not to produce all components only by themselves and IBM outsourced the production of the processor as well as the programming of the operating system. The CPU came from Intel, while the Microsoft Company, which had only 30 employees at that time, accepted a bid for the development of the operating system [cf. Gras04, 203 et seq.].

What was even more important was the decision of IBM to choose Microsoft for their operating system supplier. Although the industry was sure that the market leader Digital Research could offer a deliberative operating system, Bill Gates was chosen. The warranty of IBM that any application software has exclusively to be MS-DOS compatible helped Microsoft within the shortest time to dominate the PC-operating system market.

By the control of the operating system software Microsoft was able to establish an empire, while IBM actually lost his monopoly position. Now the market for operating systems was created because every PC producer who distributed his product with pre-installed software had to pay a license [cf. Newm06].

Generally it can be said that by the creation of industrial standards for data processing arrangements hardware became a mass product. At the same time it created the precondition for commercial software. What are, however, the specific features of this new developed product which is also called proprietary software?

Proprietary software attains its specification by the author- or copyright protection. The author grants the limited right of utilisation to the licensee, this means that the licensee can use the software copy. However, as a general rule the "legal user" does not have the right to work on the licensed software, to duplicate or to repeat. Furthermore it is not permitted for the licensee to change incorrect programme components. As proprietary software is normally expelled only in the binary code, so in the machine-readable form consisting of 0 and 1, it is also practically impossible for the user to debug. Ascertained problems of the functions can be announced to the manufacturer who is however not obliged to remove them. Only the supplier of the software has the opportunity to restore the quality of the product [cf. Lutt06 62, 80 et seqq.].

1.3 Development & History of Open Source

1.3.1 Prehistory

Even in the 60ies of the last century there were attempts to create a cross platform operating system. AT&T published the most prominent example in 1969, namely Unix. As AT&T used different architectures, the interest was extremely high to find an operating system which ran on all data processors.

For the first time it was possible to port system software in the year 1971**.** In this year the whole operating system was newly written in C. Therefore the first draft of the source code compatibility was introduced. This means that the operating system is written in the source code of a higher computer language and can be compiled to an executable machine code on the respective platform [cf. Gras04, 211 et seq.].

Thus Unix became the standard operating system of AT&T. However, at that time it was not allowed to open new business areas to the phone monopolist according to the antitrust law. Therefore Unix could not be put on the market regularly but rather be transmitted at net cost price. This happened in the case of universities but without any support. Soon the operating system reached a high level of development, because it could move many users to enhance the source code. Only the interest in reliable software of users was sufficient, without the help of a manufacturer or a developing team [cf. Ever06, 11 et seq.].

As a result of the developing activity among the Unix users of the university, who overhauled the source code, the desire for a new distribution occurred at the Berkeley University, namely the Berkeley Software Distribution (BSD). For this request it was not difficult to find a satisfying solution between the university and AT&T. The popularity of BSD also encouraged big IT companies like IBM, HP, Siemens, Sun and Novell to put their own commercial Unix versions on the market. The outcome of this was a fragmentation of the Unix market. Consequently, software producers were forced to programme different versions for every Unix mutation which lowered the attraction of the market considerably [cf. Gras04, 214 et seqq.].

While now on the one hand the commercial suppliers worked on closing the Unix codes, on the other hand especially in the university area, projects were founded to open them. Particularly BSD, which has been free software since 1992 and is totally executable, and the GNU-Project have to be mentioned.

1.3.2 Free Software

The closing of the Unix Code as well as the drastic increase of the license fees seemed to disturb only some programmers at the beginning, although they could not repair programme errors by themselves and were unable to enhance the software. The few people who had the opportunity to see the source code had to sign a non-disclosure-agreement, which means that every transmission of knowledge was prohibited [cf. Stall06a].

This situation encouraged Richard Stallman, long experienced coder with Massachusetts Institute of Technology (MIT), in 1984 to start the GNU-Project. The purpose was to create a free alternative to the Unix Operating System which should be available to everybody in the source code and which can above all be enhanced in free cooperation without being closed due to license problems. The freedom of the knowledge transfer is moved in the centre, and not the freedom concerning the costs. In Stallman's own words: "Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer" [cf. O.A.06a].

As programmers of Free Software must also live on something he enumerated a row of compensation forms in the GNU-Manifesto. In the latest 90ies this way of earning money has become accepted: Earnings income from distribution and documentation, support in terms of extending programmes but also at the beginning of the project, trainings and in matters in which software is ported on new calculator architectures [cf. O.A.06b].

From the beginning it was the purpose of the GNU-Project to offer all considerable programme functions as Free Software and to let become propriety code obsolete.

Thus becomes evident that the main goal of the GNU-Project is not to create a row of free programmes but to provide an entire system. To guarantee this, it required a juridical trick which should prevent that free software could become proprietary software.

As mentioned above, at the beginning of the computer engineering software was of no commercial value and freely circulated between developers and users [cf. O.A.06c].

In this regard one also talks about “Public Domain”. Everybody is free to act with the software whatever he wants to do with it – one can also integrate the software into a proprietary code. And exactly this is what Free Software wants to avoid. Here the General Public License (GPL) is applied. It avails oneself of the copyright, however in its reverse form. Not the claims of the license of the originator’s Free Software are regulated but it should be prevented that free code, also in changed form, can be closed or be made proprietary and that in this way the general public is not in a position to use it in a free way.

The GPL does not say, that the author has to renounce his work, but furthermore it declines which freedoms the user has. In Contrast to usual license agreements which restrict the user’s rights the GPL grants following freedoms:

- “The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this” [O.A.06a].

Software which fulfils all four criteria is Free Software. Their most significant sign is in freedom three, the so called Copyleft which prevents that Free Software is privatised and is robbed her freedoms.

Beside the GPL also the GNU Lesser General Public License (LGPL) as well as the Mozilla Public License contain such a so-called “Copyleft Clause”. The suitable regulations pose difficult questions concerning the changes of programmes. In particular, if there are changes of the programme of a secured source code or if two independent programmes are combined. While GPL and LGPL do not deliver any “clear criteria” for differentiation, the Mozilla Public License is a target on a technical differentiation sign. Provided that added code is stored in an own file, it can be said that there is no enhancement of the programme [cf. Kogl04, 301].

Till 1990 most of Unix components could be substituted with free components of the system, however the development focused more and more on user software. Only the core of the operating system, also called Kernel, could not be cancelled from the task list which main purpose was to develop a completely free operating system. Linus Torvalds, a Finnish student of informatics, who made a project at university in the year 1991, wrote a Unix-Kernel for the new Intel 386 processor which has recently come on the market. This was a great help for obtaining a completely free operating system. As the Linux-Kernel alone was of small value, the community combined it with the already existing GNU-System. The GNU-Project took on the Copyleft License in order to protect the freedom of the Kernel. After the integration of the likewise free graphic user interface XFree86, the first version of GNU/Linux appeared at the beginning of 1994. Torvalds, a well accepted authority is still enhancing the Linux-Kernel nowadays to prevent that even this project is also split in a huge number of lines as it was the case with Unix [cf. Gras04, 226ff and Stall06b].

1.3.3 Open Source Software

The philosophy of the GNU project as well as the Free Software Foundation has contributed to the motivation of the Free Software Community. Contrariwise the commercial companies were not enthusiastic about this development and expressed their negative attitude and a lack of understanding. In case of Microsoft it was not unexpected [cf. Gree06] but also the whole software sector hesitantly converged to the new operating system due to the fact of the viral Copyleft. The whole software sector was afraid of losing the control over the rights of an advancement of Free Software, as a changed programme has to be published under the same license too, under which the first version was released. Due to this procedure the expression GPL-infected arose.

However, the spread of GNU/Linux increased more and more; there was no possibility for the market to ignore that fact. At the same time protagonists of this movement stand up for a widely acceptability among companies of this new developing model. Hence, Eric S. Ramond and Bruce Perens founded the Open Source Initiative (OSI) in 1998. The main purpose was to find a more pragmatic access to Free Software Licenses and to promote the commercial use of this software. Furthermore they also wanted to replace the “communist sounding word” free by a word or item which could also be made interesting in boards or at general meetings [cf. Gras04, 230].

Now far away from Stallman’s utopia of a free software universe one tried to communicate the advantages of an open developing model to the business area. Therefore the Open SourceInitiative drew up the Open Source Definition (OSD). It is not a license itself, but it is used as a guideline for software licenses which require open source. In that way licensed software is also enabled to use the registered trade mark Open Source [cf. O.A.06d and O.A.06e].

Very much to the joy of the economy which has not been pleased about the item Free Software, as many representatives thought by mistake that it is not possible to make a profit. Furthermore the objection concerning the Copyleft does not exist anymore as the Open Source Definition does not demand that the enhanced programme must be licensed under the same license too. Contemporaneously, the most effective instrument of the GPL was given up, which should make sure that developers provide their enhancements of Free Software again for the general public for free use. Indeed, this step is also preferably seen by the representatives of the Open Source Movement and furthermore the third rule of the OSD refers explicitly to the fact that there must be the opportunity for works which are based on the Open Source Definition to be distributed under the same terms as the original software. But the uses of other licenses, in particular licenses which allow users to make software proprietary are not excluded from this rule [cf. Khar04, 7 et seq.].

2 Open Source Licenses

2.1 10 Criteria

A programme that is licensed under an Open Source License has to fulfil several criteria. Not only access to the source code, but also a compliancy with the following requirements:

1. Free Redistribution

"The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale" [O.A.06e].

Rationale: By constraining the license to require free redistribution, we eliminate the temptation to throw away many long-term gains in order to make a few short-term sales dollars. If we didn't do this, there would be lots of pressure for cooperators to defect [cf. O.A.06e].

2. Source Code

"The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost–preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed" [O.A.06e].

Rationale: We require access to un-obfuscated source code because you can't evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy [cf. O.A.06e].

3. Derived Works

“The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software” [O.A.06e].

Rationale: The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications [cf. O.A.06e].

4. Integrity of the Author's Source Code

“The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software“ [O.A.06e].

Rationale: Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations [cf. O.A.06e].

Accordingly, an open-source license must guarantee that source be readily available, but may require that it be distributed as pristine base sources plus patches. In this way, "unofficial" changes can be made available but readily distinguished from the base source [cf. O.A.06e].

5. No Discrimination against Persons or Groups

“The license must not discriminate against any person or group of persons” [O.A.06e].

Rationale: In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.

Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself [cf. O.A.06e].

6. No Discrimination against Fields of Endeavor

“The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research“ [O.A.06e].

Rationale: The major intention of this clause is to prohibit license traps that prevent Open Sourcefrom being used commercially. We want commercial users to join our community, not feel excluded from it [cf. O.A.06e].

7. Distribution of License

“The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties” [O.A.06e].

Rationale: This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement [cf. O.A.06e].

8. License Must Not Be Specific to a Product

„The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution“ [O.A.06e].

Rationale: This clause forecloses yet another class of license traps [cf. O.A.06e].

9. License Must Not Restrict Other Software

“The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software“ [O.A.06e].

Rationale: Distributors of open-source software have the right to make their own choices about their own software.

Yes, the GPL is conformant with this requirement. Software linked with GPLed libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed [cf. O.A.06e].

10. License Must Be Technology-Neutral

“No provision of the license may be predicated on any individual technology or style of interface” [O.A.06e].

Rationale: This provision is aimed specifically at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee. Provisions mandating so-called "click-wrap" may conflict with important methods of software distribution such as FTP download CD-ROM anthologies, and web mirroring; such provisions may also hinder code re-use. Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues. [cf. O.A.06e].

2.2 Categorising

In view of the rights and duties of the users the single regulations of the Free Licenses are decisive. Although all Open Source Licenses must fulfil the requirements in chapter 3.1, they differ partly considerably. A priority distinguishing feature is the difference between the so-called Copyleft and Non-Copyleft Software. The licenses can be divided according to their legal characteristics into five different groups:

- Licenses without Copyleft Effect
- Licenses with strong Copyleft Effect
- Licenses with restricted Copyleft Effect
- Licenses with restricted choice
- Licenses with privilegs

In the following subchapters the different groups of licenses are explained in more details and in every category examples are given. The categorisastion is based on the publication of the institute of legal questions of Free and Open Source Software [cf. ifro06].

2.2.1 Licenses without Copyleft Effect

Licenses without Copyleft Effect are characterised that they grant all rights and privileges of an Open Source License to the user and that they do not restrict the user in case of modifications with further conditions. Subsequently, the respective licensee may distribute modified or enhanced versions of the software under a license of his own choice. Therefore the licensee can transform Free Software into proprietrary software [cf. ifro06].

Examples for this license group are:

- BSD License
- Apache License
- MIT License

2.2.2 Licenses with strong Copyleft Effect

Licenses have a strong Copyleft Effect if the licensee has to publish the modified software only under the same license terms as the software originally acquired. This, however, does not necessarily mean that the licenses as mentioned below are “GPL compatible” [cf. ifro06].

The Copyleft was invented by Richard Stallman who said: “To Copyleft a program, first we copyright it; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code or any program derived from it but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable” [O.A.06j].

Examples for this license group are:

- GNU General Public License
- Common Public License
- IBM Public License
- Open Software License

2.2.3 Licenses with restricted Copyleft Effect

Licenses with limited Copyleft Effect are similar to the licenses with strong Copyleft Effect, because they have a Copyleft Effect too. Here, however, the Copyleft Effect is limited. If modifications are made in a separate file, such file’s content may be distributed under a license which is different from the original license, for example a proprietary license. Under these types of licenses it is possible to combine software under different licenses more easily [cf. ifro06].

Examples for this license group are:

- Mozilla Public License
- GNU Lesser General Public License
- Sun Public License

2.2.4 Licenses with Restricted Choice

Licenses with restricted choise have a number of legal effects which depend on the scope of the respective modification. Furthermore, the licensee has a restricted choice of how to distribute modifications of the software [cf. ifro06].

Example for this license group is:

- Artistic License

2.2.5 Licenses with Privilegs

Licenses with privileges grant to the licensee the entire rights which are characteristic for Free Software, but at the same time they reserve specific privilegs for the licensor, in case the licensee changes the software. This kind of license is often used turning proprietary software into Free Software [cf. ifro06].

Examples for this license group are:

- Netscape Public License
- Apple Public Source License

2.3 License Models

According to the illustration 1 it can be mentioned that the predominantly Open Source License which is used in practice is the GNU General Public License. In the year 2003 there were nearly 3/4 of all projects, listed on the Open Source Knowledge Data Base SourceForge.net, licensed under GPL [cf. Foku06].

According to this the most used licenses will be explained in this paper.

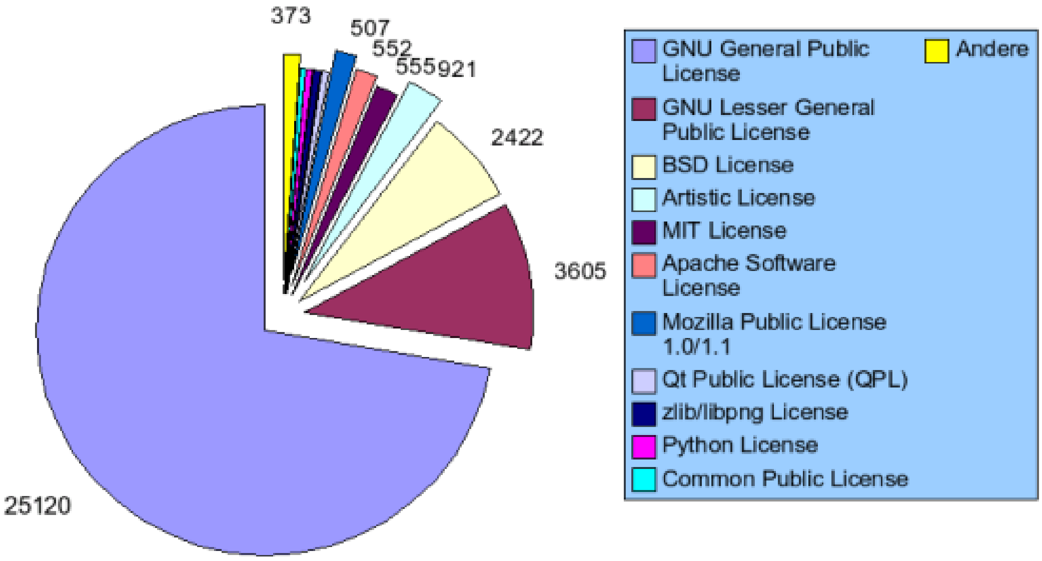


Illustration 1: Allocation of Open Source Licenses [Foku06]

2.3.1 GNU General Public License

The GNU General Public License (GPL) is the most important and widely spread license for Free Software. At the same time it is the flagship of those licenses with a strict Copyleft Effect. Among other things the Linux-Kernel and the GNU Components (Emacs, GNU Compiler et cetera) belong to the GPL. [cf. Quad06].

Rights

The licensee receives the rights to modify, to copy and to distribute the software. Besides, the rights to distribute cover the passing on of the software on data carriers, as well as supplying the software for downloading on the Internet.

The GPL entitles the rights mentioned above to users directly. This means that the rights are directly provided by the author and not by the distributor of a perhaps commercially distributed version [cf. O.A.06k].

The GPL is to be seen as an offer to everyone for a conclusion of a contract which provides the above mentioned rights of use. As soon as a user changes or distributes the software, he accepts this offer [cf. O.A.06k].

Seen from copyright points of view the competence of the users to change licensed software under GPL seems to be a problematic issue. According to the copyright law the originator is entitled to forbid adverse effects or deformation of his work in case that personal or spiritual interest is endangered. Especially, as the programmers of Free Software are not primarily interested in commercial success, but in the development of a good reputation within the community. In exceptional cases it may cause infringement of the copyright in spite of the permission to change software and the duty to mark changes. For further details see chapter 2.4 [cf. O.A.06k].

Liabilities

Like all Open Source Licenses GPL entitles not only the rights of use, but they all combine the entitlement with duties. These duties differ dependent on the fact whether an unchanged or changed version of the software is distributed and furthermore whether a distribution is carried out in source code or object code. The pure internal adaption of the software without a distribution is not linked to special obligations.

Section 2b contains the basic rules of all Copyleft Licenses: Derivations of GPL-Software are only allowed to be distributed or published as a whole under GPL.

In principle section 4 of GPL contains the most significant instructions concerning the duties of the user: “You may not copy, modify, sublicense, or distribute the Prgoram except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License” [GPL section 4]. This means that any disregard of the licensing requirements causes a termination of the contract. Afterwards the user must not use, work on or distribute the software.

In the following subsections the specific duties of the user of GPL Software will be described in detail.

Distribution of Non-Modified Versions

With the distribution of non-modified versions the licensee must attach a copyright notice as well as guarantee exclusion and a disclaimer of warranty. Already existing notes concerning the license respectively missing liabilities must persist and a copy of the text of the license must be distributed, too [cf. Nütt06].

The attachment of the license text is of essential meaning, because otherwise third parties cannot make use of their rights. There must always be a clear classification of the software; this means it must be clear which license is valid for which part of the software [cf. Nütt06].

In case the software is only transmitted in the object code, one should pay attention that the source code of the software is available as well. In this way the possibility is created that the software can be really changed. The GPL allots four different ways:

- Direct: By means of a common data medium (CD-ROM).
- Indirect: A written offer (valid for three years at least) to everybody concerning the consignement of a complete source code which will be delivered on a common data medium (only costs for the physically source distribution are allowed).
- In case that the distribution occurs in a non-commercialized way and the programme has been aquired without any source code and just only by means of a written offer, it is sufficient that this offer is distributed to the purchaser again.
- If the programme is distributed via Internet the source code should be offered on the same website for downloading [cf. GPL section 3].

Distribution of Modified Versions

Due to the modification of the version further obligations arise in addition to the already mentioned duties above, if the changed software should be distributed.

Modified versions need a remarkable note (with information about the way and date of modification). This marking is also possible in anonymous or pseudonymous form. It is sufficient if the endorsement of the modification is included in the source code, because everybody who transmits the software in the object code is at least obliged to offer the source code. In this way it is made sure that the consignee has the possibility to take note of the endorsement of the modification [cf. O.A.06k].

As crucial point of the GPL the Copyleft must be respected: Anybody who changes the software or a part of it and distributes or publishes the modified programme is obliged to license the software under GPL according to the rules of section 2b: “You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License” [GPL 2b].

The principle is easy: Additional or other terms of licenses are not allowed. The transmission may only occur under GPL. This arrangement is starting point of many discussions. On the one hand it is necessary to find an extensive solution which also protects Free Software in the future against being used as proprietary software all of a sudden, and which is therefore no longer freely accessible for all as Open Source Software [cf. Nütt06 and GNU 2b].

Combination with Components of Software under other Licenses

When you want to combine GPL Software with other licenses the same rule is applied like for the addition of new code components: A transmission has to be carried out under the conditions of GPL, as soon as the outcome contains components of GPL Software or is derived from it. If in contrast several independent programmes are concerned the original GPL Programme has still to be transmitted under GPL, however, the combined programmes can be delivered under their own licenses (proprietary or free) [cf. O.A.06k].

The decisive criteria whether the combination of software components has to be licensed under GPL too or not underlies the term “Derivative Work”. If added software components don’t represent such a ”Derivative Work” together with the GPL-Code, the former can be transmitted under other licenses without infringing upon the license regulations of GPL [cf. Gras04, 284 et seqq.].

According to section 2 article 2 GPL one can speak about "Derivative Work" if identified segments of the work are not derived from the programme and reasonably can be regarded as independent and discrete compontents [cf. GPL 2].

GPL goes pretty much into detail and tries to cover all kinds of potentialities. According to CPL it is strictly arranged so that it is possible to check whether a programme is a derived work or not. (Here the literature tries to define technically about: Corporate loading, dynamic or static linking, executables, embedded systems etc.) Particularly, just the use of libraries is handled very strictly and hardly to elude, hence “Library GPL “(nowadays Lesser GPL, shortly: LGPL) was introduced by the Free Software Foundation. The following chapter dwells on LGPL [cf. Gras04, 284 et seqq.].

To sum up it may be said that GPL deals severely with Copyleft in order to guarantee the protection of the Free Software in the long run.

2.3.2 GNU Lesser General Public License (LGPL)

The LGPL is an enlargement of the GPL which was born from the need when one recognised that through the development and distribution of Open Source Software questions and problems arise as mentioned above when using programme libraries. Libraries are collections of programme functions and data which are used for the development of software. Libraries differ from executive programmes inasmuch, as they are not independent programmes, but they offer services to other independent programmes as a kind of help [cf. Brok06 and O.A.06k].

In the preamble of the LGPL the reason for a reduced protection of the freedom of the software is stated. The reason for problems of libraries is that it is often difficult to distinguish between the use of software or the modification. Legally speaking it can be said that the linking or the combination of an application with a library licensed under GPL leads to a derivated work which means that the same application would have to be licensed again under the GPL in the practice. Thus it is neary impossible (even in the case of pure combining of an application with a library or during the use of compilers under GPL) to elude the GPL which often put off many programmers and highly influenced the development of the Open Source Movement [cf. Brok06 and O.A.06k].

Libraries which are licensed unter the LGPL allow therefore also a use with differently licensed or even proprietary software. After achieving sustained success with the distribution of Open Source Libraries – at the expense of the originally proclaimes freedom – it was quickly critisiesed by the group of Open Source Hardliner headed by Stallman that now the basic principles of Copyleft have been broken (also by commercial providers). In this regard the originally as "Library GPL" known license was renamed to the now well known "Lesser GPL". One distanced oneself from the term "Library" to the meaning "less" (worth) than the GPL with their strict occurrence of the Copyleft. Moreover the Free Software Foundation wanted to refer to libraries not having to be licensed exclusively under the LGPL. In contrast to it is emphatically suggested also to put libraries again under the GPL. The LGPL is supposed to be used only then if it is essential to establish a new library as a standard of the market [cf. Brok06 and O.A.06k].

Basically the GPL and the LGPL only distinguish technically in the way of linking programme libraries with an application. However, it remains a license with Copyleft Effect.

2.3.3 Berkeley Software Distribution License (BSD)

The BSD License was developed by the Berkeley University. This license does not plan a publication of the source code compulsory so that it is also possible to publish only binary files. Besides, derived products must not be subjected mandatorily again to the same license. Therefore it is possible to use a BSD Software in commercial products. Consequently, software which is published under this license is nearly not subjected to restrictive conditions, which can be traced back to the historical development [cf. Renn06, 21]

The first BSD License was founded in the year 1989. In the meantime different versions of this license are available. Furthermore it has sampled as a model for numerous similar software licenses. For example: Apache License, X Window System License, W3C Software Notice License and the OpenLDAP Public License [cf. Brok06, 7 et seq. and Gras04, 279 et seqq.].

Rights

As already mentioned above, at all Open Source Licenses specific usage rights are issued to the licensee. He may duplicate, change and publish changed and unchanged versions off- and online. BSD specific special features do not exist [cf. Gras04, 279 et seqq.].

Liabilities

The central duties of the licensee differ from the user's intentions whether he only wants to distribute unchanged copies of the software either in object code or source code or if he has the intention to modify the software and afterwards to distribute these programmes.

Distribution of Unmodified Versions

For the distribution of unchanged versions of programmes the BSD Licenses make regularly arrangements like all Free Software Licenses which are primarly for the purpose to entitle the aquirer of the copy of the programme the possibility to profit from the Open Source Model.

The distribution of unmodified software copies of programmes is linked to the user's obligation. So the licenser has to offer the licensee of the software a copy of the Open Souce License under which the software is distributed. In this way every acquirer should have the possibility to obtain the author's rights to copy, modify and distribute the programme by accepting the license agreement [cf. Brok06 and Gras04, 279 et seqq.].

In addition to the obligation for transmitting the license agreement the BSD Licenses plan ahead that the duplication of the source code has to contain the copyright notice which refers to the author. According to section 1 of the BSDD License it is necessary to mention the exclusion of liability of the author of the software [cf. Brok06 and Gras04, 279 et seqq.].

Modifcation of the Software/Distribution

Naturally, it belongs to the concept of the Open Source Model that the modification of the software for own purposes is possible to everyone without restrictions. The pure internal processing is therefore not linked to any duties. There are not any duties to be recognised as long as the modification of the software is for own purposes, but they have to be recognised at the moment of distribution or publishing of the software.

According to the copyright law it can be said that there is no personally relatedness among the members of a company, but a professional cooperation, which means that in case of a distribution with BSD Software in a company the obligations of the BSD have to be respected.

But it has to be mentioned that in case of a pure test or modification of the software within a manageable development team no contractual duties arise.

Different from licenses with a strict Copyleft Effect the BSD License does not plan any duties for the disclosure of the modified soure code, in case that the modified version should be distributed. The licensee should be able to decide by himself in which way and volume the source code should be published and to which extent he grants third usage rights at his own copyright. In particular he can distribute and publish modified software in a proprietary way [cf. Brok06 and Gras04].

In this regard the wide difference between the strict Copyleft Licenses and the BSD License becomes apparent.

However, corresponding obligatons have to be observed which are also valid for the distribution of unchanged versions as already mentioned above. A copyright notice, the license agreement and a disclaimer of liability have to be added to the programme.

Furthermore older versions of the BSD contain an endorsement of advertising which compels that derivates of the software must contain the name of the University of California with reference to the Lawrence Berkley Laboratory. This advertisement does no longer exist with the newer versions [cf. Brok06 and Gras04].

Series of BSD License versions contain a clause which prohibits the nomination of the original author of the computer programme as long as no explicit written allowance is given. Thereby the distribution of modifications should be promoted with the purpose that after having enforced qualitative bad changes of the software by a third party no advertisement can be made with the original author’s name. So in this way it is guaranteed that the original author does not appear in a bad light.

2.3.4 Apache License

The first version of the Apache License was published in 1995. Since January 2004 the current version 2.0 of the Apache License is used.

In the older versions (1.0) the Apache License contained an endorsement of advertising which structor is similar to the former BSD License. The copyright of third parties, who have contributed to Apache, have been transmitted to the Apache Software Foundation according to the advertising. To avoid that the copyright is transmitted to the Apache Software Foundation the current Apache versions are now always available without an endorsement of advertising.

Also this license entitles the licensee to modify and distribute the software. However, the obligation that the source code has to be transmitted is not granted like it is the case with the BSD License. Furthermore the software is not liable to the Copyleft Effect, so derivates may not be published under the same license as the original document. Thus the operators have the privilege to transmit the Free Software in proprietary one [cf. Broc06].

Advocates of the BSD similar Licenses do not have the same philosophy as those representatives of the GNU Licenses. They do not regard their license model for a political media to make software available for the general public. As the interests are pragmatic by nature the Open Source Model can be seen as one of the most efficient ways to develop good software. This method keeps the developers from the necessity to programme a redundant code. If the problem is solved once, the solution is made public and other developers save time and work and can use the code immediately [cf. Broc06].

This license does not conatin obligations like in contrast to the licenses of the Free Software Foundation to which licensees are obliged. Yet, the usage of the disclaimer of liability and the obligation to use the copyright notice are restrictions which have to be obeyed [cf. Broc06].

A special feature of the Apache License is that the distribution of modified programmes under the same original name is only allowed after having received the prior authorisation of the initial author. This should prevent that a mix-up of programmes occur between the original version and their derivatives. Furthermore the derivative must allude to the fact that this programme is based on software which is published under the Apache License.

The usage of the Apache License and other BSD similar Licenses bring about that no other products licensed under GPL can be integrated, as this would infringe upon the procedure of the Copyleft. On the other hand products which are licensed under Apache or other BSD similar Licenses can be integrated in GPL Programmes without getting in the way of restrictions of the license agreement [cf. Broc06].

2.3.5 Mozilla Public License and NPL

The Mozilla Public License (MPL) allegorises a balance between the Copyleft and the BSD similar Licenses. On the one hand it stipulates that modifications which are based on MPL Software have to be licensed again under MPL, but on the other hand it is less strict than classical Copyleft Licenses concerning the corporate handling with proprietary programmes.

In 1998 the source code of the Netscape Navigator, which was distributed until this time as proprietary software, was disclosed. This measure was taken in order not to lose ground against the increasing pressure on the part of Microsoft which had the browser “Explorer” on the market. The license regulations were edited to this case in particular. In order to sustain the operativeness it was allowed on the one hand to distribute the software with proprietary software (therefore the GPL dropped out as a possible license). On the other hand a BSD License could not be chosen due to the intention not to lose the control of the development of the browser [cf. Gras04].

Thus the Copyleft is only valid for modifactions which are not published in own files. Therefore such changes must subject again to the MPL. If changes are stored in own files, that is to say from the initial developer, they are not affected by the Copyleft. According to the section 13 it is allowed that the initial author of the software can license the code under MPL as well at the same time under an alternative license, too. However, this rule is not valid for contributors [cf. Gras04, 298].

The first version of the license agreement seemed to be have written by copyright-lawyers. For the first time a Free License dwells on possible patent claims. Due to vehement public discussions and criticism against the special rights which have been reserved by Netscapae, the company decided on publishing two licenses. The Netscape Public License 1.0 (NPL) and as well as the Mozilla Public License 1.0 (MPL). The NPL was written in purpose of the released code of the Communicator and for all derivated works. The MPL License can be used by authors, who do not want to give Netscape the possibility for a privileged access, for independent works [cf. Gras04, 307 et seq.].

Both licenses consist of an identical main part. Additions are only added to the NPL to control the special status of Netscape [cf. Gras04, 308].

The MPL also grants all freedoms of Open Source Software and demands that all modifications are made accessible to source code form and that those are published under the same license (section 3.2). However, section 3.6 accords to publish the changed or unchanged software exclusively in object code under any different license, provided that a reference to the free source code is added [cf. Gras04, 308].

The main focus it to be put on section 3.7. Here the user is allowed to circulate MPL licensed software with differently licensed software as a whole. At this point the MPL considerably differs from the GPL at which it is only allowed to distribute programmes as a whole in case all parts are taken under the GPL. Consequently, the linking to a Larger Work of MPL Code with code under a different license is according to section 3.7 possible. Such a Larger Work is not interpreted as Derivated Work as long as the initial source code keeps on being controlled by NPL or MPL [cf. Gras04, 308].

Here the starting point is the question, whether the new code can be saved in a new file (then no releasing duty exists) or the enhancement is added to the existing source code. To result from this – provided that it it technically possible – enhancements, which can be written and saved as a standalone in a new file, can also be offered under proprietary software. In spite of that the source code has to be easily accessible on the same storage medium or it must be possible to keep the original code regardless of which kind of publishment (commercial or not) ready. All changes have to be sufficiently documented and a copy of the license must be added [cf. Gras04, 308].

Thus for example it is conceivable that an enterprise like Microsoft enhances the free code by attractive functionalities and distributes the entire programme exclusively in proprietary form. If enough users of these supplementary functions are disposed to give up the freedom, developers of the Free Software will lose their operators and the free project will be doomed to failure [cf. Gras04, 308].

Thus the NPL and MPL carry out an unusual seperation between the original author (for the NPL code it is Netscape and for a not derivated independent work it is the author who underlies the software under the MPL) and the contributors (section 2.2). In the amendments of the NPL a further distinction is carried out between the version of the Communicator under the branded version Netscape and the free versions under the project name Mozilla. The sections which allow Netscape, to use NPL-Code including the modifications carried out by a third party in its branded code (section V.3) and in over two years after the release of Mozilla to use in different products (section V.2) without being bounded to its own license, are controvers. Furthermore Netscape reserves to license code under the NPL under other conditions than the NL onto a third party. These amendments annul effectively the rules of freedom in the main text of the NPL for the company Netscape (section V.1) [cf. Gras04, 308 et seq.].

The reasons for these exceptions are definded on the part of Netscape that the fractional used code for the client “Communicator” is used in server products too. Thus it should be guaranteed that changes in the server code and modifications of free developers could be integrated in proprietary code without having to subject these modifications likewise to the NPL [cf. Gras04, 309].

Similarly to the GPL it also leads in the case of noncompliance of the MPL License agreements to a cancellation of the contract. However, a longer time limit is set (30 days).

2.3.6 Artistic License

The Artistic License is used for the programming language Perl. Perl itself is also licensesd parallelly under the GPL; the same is also valid for many Perl progammes and libraries.

The Artisitc License permits free distribution or transmittance of modified programme code but it draws a distinction between the so-called standard-version and its derivatives of it. It asserts a claim to produce proprietary derivatives which have to be explicitly labelled (other file names, documentation of the differences).The free appropriation of the source code concerning the own modifications can take place once for example via posting in Usenet or via an upload in a public archiv for data. A charge for distribution may be demanded (so fees for the pure appropriation of the software) but no license fee is allowed (so fees which are linked to the number of the users) [cf. Artisitc].

2.3.7 Common Public License

In the year 1999 the Common Public License was published by IBM under the name IBM Public License. In 2002 the new version 1.0 was created, but in spite of the new name CPL is basically identical with the IBM Public License of 1999. The new name facilitates the use for other companies. The most important software which is licensed under CPL is the IDE Eclipse [cf. Boek06].

As it is the case with other licenses one can use software which is licensed under CPL freely, furthermore one has the right to modify and to distribute the product. In addition the CPL grants a license of patent too. Even though a developer enhances an Eclipse Module and patents it for example, it would not bring about any problems as the enhancement is also licensed under CPL and so everyone receives a patent license.

Therefore the CPL is not accredited by the Free Software Foundation, but of course by the Open Source Initiative [cf. Boek06].

According to section 2 CPL each user of the licensed software may reproduce, prepare derivative works, publicly display, publicly perform, distribute and sublicense the contribution in source code and object code form [cf. Boek06 and CPL].

In section 1 (b, ii) there is declared that one can add separate modules which are licensed under another different one. In this way one wants to encourage other enterprises to develop proprietary modules which can be added to Eclipse [cf. Boek06 and CPL].

One cannot denote a module as spearate one, if the module is a derivative work. Is therefore the newly developed module a derivation of a CPL Software, it cannot be published under a proprietary license [cf. Boek06].

According to section 4 CPL commercial distributions are permitted, but several specified things have to be considered. As aforementioned the problem about derivative works can be a complex matter which often demands a legal consultation. Furthermore if CPL licensed software is sold the seller is subjected to liabilities [cf. Boek06 and CPL].

2.4 GPL in Connexion with the European Law

The GPL has been edited for the American field of law as well as the most other licenses, too. Stallman namely emphasises that the licenses are based on the Bernese Agreement concerning the copyright law, however, it should be checked whether it represents a valid license agreement according to the European fields of law [cf. Gras04, 286].

Due to the many guidelines on the part of the EU the European copyright is harmonized to a large extent. Consequently, it can be emanated from the assumption that the copyright law is nearly identical with the Austrian one. According to the “European Community Directive on the legal protection of computer programmes (91/250/EEC)” software is protected as literary works in case of the copyright law [cf. O.A.06n].

According to the Austrian copyright law an author is entitled the exclusive right to his work (§ 14 UrhG). Consequently, he can decide on his right of reproduction (§ 15 UrhG) and his right of distribution (§ 16 UrhG). The crucial points of the GPL may be conferred on the Austrian Copyright Law. The author can dispose of the way how his work is used by other users. This can include one or more of the following rights: The right of reproduction, distribution and modification [cf. Gras04, 286].

The implicit mechanism of the section 5 GPL enables that a valid contract between the author and the user of the software is accomplished. The user accepts impliedly the license agreement by using the software whereas a notification to the author of this acceptance is not needed. The linking of the usage rights to certain conditions is also legally, for example to make the source code available or derivated works have also to be published under the same license again [cf. Gras04, 286 et seq.].

The greatest differences between Open Source Licenses and copyright concern the intellecutal property rights. While the right of the author to mention his name is guaranteed (according to the obligations to retain the copyright notice as per section 1 GPL as well as the declaration of modifications as per section 2a GPL), the protection of integrity can cause some problems. Conceivable problems could arise so that the author may prohibit the distribution of the software if the modification of the programme provokes any defamation. In accordance with the Austrian (§ 21 (3) UrhG) and the German law (§ 14 UrhG) the author may prohibit the derivated software. Based on the copyright law it can lead to a prohibition brought about by the auhor in an exceptional case, although the GPL grants an extensive release of the Open Source Software. Therefore an element of risk remains [cf. Gras04, 287].

Warranty

The warranty affects section 11 of the GPL and completely excludes these, anyway "so far permissible by law". With the contract between the user and the distributor of the software (this means the GPL License Agreement) the protective promision should be disabled. The trouble is that the GPL represents a pre-formulated general term and condition. The EU directive "93/13/EEC of 5th April 1993 on unfair terms in comsumer contracts" arranges that one can shirk responsibility and to assume no liability by means of “small print” [cf O.A.06o]. Thereby the European directive does not allow a reduction of warranty which is constitutional in contrary to the US Law. This means that with the verbalisation “to the extent permitted by applicable law” it is not possible to cut the general terms and conditions. Hence, the general legal requirements are valid, if the clauses contradict the legal requirements – and this is the case with section 11 of the GPL. For a complete caveat emptor is ineffective. Therefore section 11 GPL is as if there was no exclusion of warranty. However, this does not mean any exclusion criteria for the refusal of a warranty [cf. Hohe04, 19].

In a normal case in which software is offered for downloading at no charge under an Open Source License one can act on the assumption that it is a donation. Not only things can namely be donated but also other miscellaneous services. The usage right of a programme is such an allowance. The person who is donating an item should not be responsible for smaller problems but only for deficiencies covered with malice aforethought or guilefully unmentioned. Therefore that person who does not know that his programme is inoperable does not have to vouch for occurring shortcomings [cf. Hohe04, 19].

However, the state of affairs looks more complicated for distributors: If they sell a proramme on CD-ROM and with manual or support a product to a flat rate, a warranty according to the commercial law is conceivable. A salesperson is always subjected to stricter conditions than someone else who does something for the public or anyone free of charge [cf. Hohe04, 19 et seq.].

Exclusion of Liability

The same is valid for the exclusion of liability according to section 12 GPL. It is also ineffective, but fortunately the legal regulations for the donor are also gentle. The person who is donating has only to take the responsibility for intention and gross negligence. Gross negligence occurs only then, for example, if someone offers a hastily joined programme without any further examination and thereby inflicts damage to hard- and software [cf. Hohe04, 20].

In the meantime the GPL has passed their juridical “baptism of fire”. In May 2004 the regional civil court for Munich confirmed the temporary act of disposal against a tradesman who used GPL licensed software without adherence to the license agreement. Accoding to the proclamation of sentence, the judges honour the GPL fundamentally as effective in law. They consider the complaining developers of the software authorised and so they can assert the copyright on the source code [cf. O.A.06l, Land06, and Widm06, 168].

2.5 Outlook on GPL Version 3

After 15 years a new version of GNU General Public License has come up for discussion by the Free Software Foundation. Nowadays numerous enterprises have economic interests in the details of the license, simply because they either distribute software under GPL or use GPL Programmes. The following main topics are being discussed at the moment [cf. Jaeg06, 46].

Digital Rights Management

An important change in the GPLv3 concerns the issue Digital Rights Management (DRM) – in the draft it is mentioned as Digital Restrictions Management. Basically, one is of the opinion that DRM is incompatible with GPL whose aim is to protect the users’ freedom. As an example it is an offence against GPL if a programme is changed in such a way that it interferes with the private sphere of the user which is prohibted according to the applicable national right [cf. Jaeg06, 46].

Attention should be paid to the regulation that GPL Software must not be distributed together with a DRM system in a way that the users’ rights concerning the programme become restricted. Consequently, GPLv3 prohibits de facto any distribution of GPL Software which is provided with DRM-Systems in order to regulate the use of the software [cf. Jaeg06, 46].

The second paragraph of number 3 deals with the use of GPL Software to restrict the access to other products, in other words this paragraph deals with the copyrights. In the text of the license it is literally said: “No covered work constitutes part of an effective technological protection measure” [GPLv3]. Whether a technique is an effective protective measure or not can only be answered by means of the originator’s law. GPL cannot exert influence on this fact [cf. Jaeg06, 46].

Patents

Innovations arise in view of software patents. Section 11 article 1 clarifies that the Copyleft also refers to any patents. Therefore the new version follows the example of the Mozilla Public License as well as the Apache License which already provide an explicit inclusion of patents: Someone who distributes GPL Programmes under the new license version must grant a simple license without a special commission to all patents which could be violated using the programme [cf. Jaeg06, 46].

Section 2 may cause further discussions, because it obliges the Distributor of GPL Software to protect his customers against existing patent claims of which he knows. Therefore GPLv3 would like to prohibit that distributors leave buyers out in the rain [cf. Jaeg06, 46].

Compatibilities

In the recent years incompatibility has turned out to be a practical problem with the use of Open Source Licenses. To escape from this dilemma the new figure 7 of the GPLv3 draft permits the combination with code under licenses which concede further rights to the user [cf. Jaeg06, 48].

International Law

In some European states legal analyses have shown that the liability and guarantee exclusions which are effective in the USA are effectless in other legal systems (see chapter 2.4). Besides the dimension of the usage rights can be mistakeable because of the different choice of words in the different copyright laws [cf. Jaeg06, 48].

The GPLv3 draft attempts to solve some problems in the international legal relations. So the new item “propagate” should cover all acts of utilisation, independent from the national copyright of the different countries, for which a permission of the author is required. It will have to be investigated, whether it is possible to sort out effectively the problem concerning the construction of the license [cf. Jaeg06, 48].

Implementation

At an infringement of the license obligation the injured person loses all the rights given by the GPL, so that he can be seen as a piracy. In Germany this strict regulation has already led to the enforcement of the GPL several times. Now this rule is supposed to be replaced by a right of chancelation, whereby one wants to prevent that a user loses his usage rights immediately in case of unintentional license infringement [cf. Jaeg06, 48 et seq.].

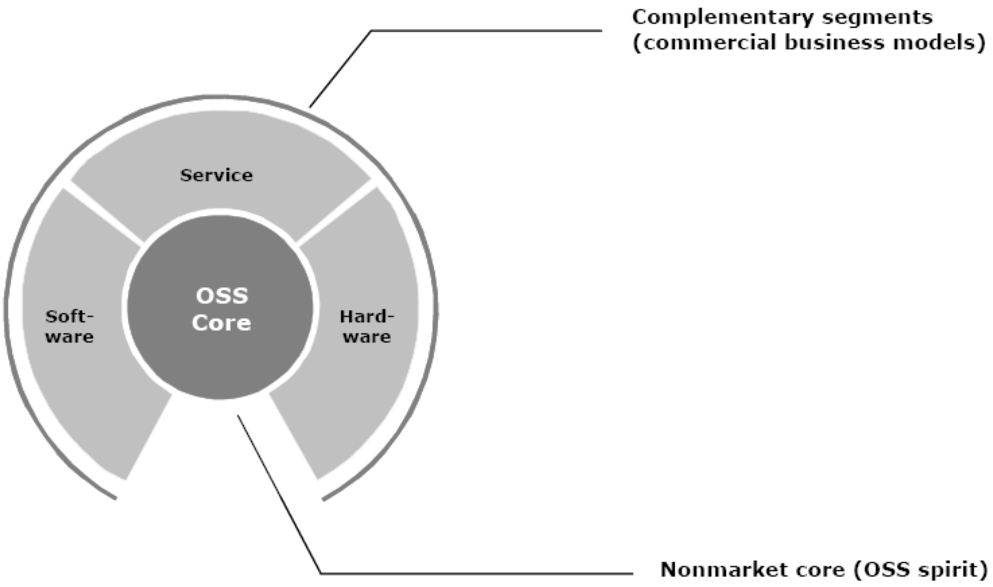
If the new version is finished, one will receive the free decision-making authorithy under which version of license he would like to license his software. It is already possible in the current version of the GPL (according to section 9 GPLv2) that the licenser can decide under which license version he would like to license his software [cf. Jaeg06, 49].

3 Open Source and Economy

3.1 OSS as a Basis for Business Models

Due to the fact that Open Source Software may be legally copied and for example downloaded from the Internet excludes the commercial use of software by using cost-based licensing models. Accordingly, there is no market [cf. Koot06, 41].

Nevertheless, business models are existing which are based on Open Source Software. Those models build on Open Source Software and use it for their own complementary products. Companies which are involved in Open Source Projects try to improve these with the purpose of gaining profit through their additional products with costs. As improvemetns in Open Source Software cannot be used directly to generate profit, companies have to benefit from indirect products and services. As sales are not brought into being with the actual product but with additional products and services based on open source, Open Source Business models can therefore be described as indirect business models [cf. Koot06, 41 et seq.].



.

Illustration 2: The Open Source Core and Indirect Open Source Business Models [Koot06, 42]

It has to be mentioned that companies – contrary to the developer community – have primarily financial motives for participating in Open Source Projects. The illustration 2 points out different ways of designing commercial business models which are based on open source. One distinguishes between business models offering additional services, additional software and/or additional hardware [cf. Koot06, 42 et seq.].

3.1.1 Selling Additional Services

Additional services are offered by distributors of Open Source Software like for example bundling, testing and adaption of that software. There are different well known distributors for Linux like Red Hat, SuSE or MandrakeSoft which give the end user the opportunity to purchase different software bundles for various purposes (for example server applicactions, desktop applications, software for administrators or developers) [cf. Koot06, 43].

The advantatges for a buyer of these bundled software packages are that the user no longer needs to search for the software, download and then adapt it so that the individual components faultlessly operate together. Additionally, any buyer has the opportunity to obtain upadates or bugfixes in particular sequences. As a matter of course, any user could also bundle and install these components and fix any bugs autonomously [cf. Koot06, 43].

The distributors of a company engage their own developers who adjust the software as required. It has to be stated that every development contribution which was designed by a distributor to his own Linux distribution must, however, enable a free access [cf. Koot06, 43 et seq.].

A big drawback for distributors is the minor leeway in pricing their products. Users can also bundle their products themselves for free without laying claim to any distributors. A further disadvantage is that users can pass on distributions free of charge. Moreover, other companies can take over a given distribution and allocate it as their own product [cf. Koot06, 44].

Due to the fact that the restricted opportunities to generate profit wtih the sale of distributions, many distributors also offer consulting, implemtation and training services.

The aquired know-how of bundling software components may be helpful [cf. Koot06, 44].

3.1.2 Selling Additional Software

To generate profit through complementary proprietay software an Open Source Base is the pre-condition. In the normal case these companies have a very close relationship with individual Open Source Projects. Furthermore, in many cases the company founders are also the initiator or supporters of the projects [cf. Koot06, 44 et seq.].

The main idea of these business models is freely available Open Source Software on which additional add-ons or programmes with enhanced functionality are built and consequently distributed for fees. In addition, the dual-licensing strategy enables that a product is freely available as Open Source Software and the very same product can also be sold as a cost-based version. Maybe, one reason for this is that companies want to use it in combination with other proprietary software. Another argument could be that someone is thrilled of this add-on and is therefore willing to pay [cf. Koot06, 45].

Nevertheless, these add-ons or programmes with enhanced functionality are mostly dependent on the support of the Open Source Community [cf. Koot06, 45].

3.1.3 Selling Additonal Hardware

Hardware can also be a promoting tool to increase sales combined with Open Source Software. IBM can be mentioned as an example. Until 2001 IBM has invested 1 billion US dollars in different Open Source Projects which included adapting Linux and Apache to the different IBM hardware platforms [cf. Koot06, 46].

The ulterior motive is, that then software is freely available for hardware. So no any extra costs for purchasing operating systems incur. If the user has fewer software expenses, however, the hardware firm’s pricing leverage increases [cf. Koot06, 46].

3.2 OSS in Practice

3.2.1 Consideration of Profitability – Study by Fraunhofer-Gesellschaft

In the year 2005 the Fraunhofer-Gesellschaft carried out a study about the cost effectiveness of Open Source Software. The concrete szenario is the migration of personal computers with Microsoft-Software into Open Source Software. The calculation is based on a total cost of ownership analysis and all cost drivers are determined and valued monetarily [cf. Frau06, 154].

The outcome of the study was a reduction of costs of 2.4 percent at the comparison of full costs. However, when regarding only the costs which come into existence because of the considering intention, the economic potential is 6.9 percent [cf. Frau06, 154].

By comparing the full costs one finds out that there is a reduction of 7 percent concerning the license fees. However, the additional expenses for training are about 2.7 percent higher at the beginning of the migration. Especially, the administration is burdened with doubled time and effort. During the adoption of Open Source Software the productivity of the employees is little and therefore the costs are about 0.7 percent higher [cf. Frau06, 166].

A comparison with the Soreon-Survey shows that there are some differences. A medium-sized company (about 100 personal computers) can save about 6 percent and a big company (about 2000 personal computers) can save about 20 percent. The differences between these results are due to the different sizes of the companies. Further criteria are for example good knowlegde of information technology, which is available for scientific employees [cf. Frau06, 167 et seq.].

3.2.2 Results of an OSS Study in Vienna

In these days not only companies are interested in Open Source Products, but also the public authorities. The municipality Vienna has already been using a server-system with Open Source Software with great success for many years. At present they think of introducing Open Source Software on personal computers and therefore they had carried out a study [cf. Magi06, 31].

The result of that was that actually 7500 out of 16000 personal computers can be converted easily into OpenOffice.org (4800 out of 7500 are also appropriate for Linux). One does not intend to convert the other personal computers because of functional and economical reasons. Therefore the municipality Vienna aims at a long-term coexistence which, however, causes technical and organisational requirements [cf. Magi06, 31].

Furthermore, the study points out that all users are dependent on the information technology of several firms under the current market conditions. Therefore there are lots of negative consequences for some companies and operators like the municipality Vienna [cf. Magi06, 31]:

- Huge producers of software are widely autonomous in their license and price policy.
- The decision-making process concerning technology is restrictively influenceable.
- The security of the products is not checkable and because of monocultures addtionoal risks of security arise.
- Maintenance and bugfixing can only be carried out by the developer [cf. Magi06, 31].

Contrary to proprietary software Open Source Software makes a contribution

- to reduce the dependence of external developers of software,
- to prevent that several providers can use their dominating position on the market,
- to obtain a better quality due to a cooperative development process, and

- to ease the internal and external data exchange via the use of open and standardised interfaces [cf. Magi06, 32].

Nevertheless, the usage of Open Source Software is risky. On the one hand the coexistence of two systems takes a technical risk. On the other hand an important factor is the acceptance of the employees, who have to be led through supporting measures [cf. Magi06, 32].

3.3 Economic Motives for Participation in OS Projects

Generally, there are some prejudices concerning the motivation of working for Open Source Projects, one of them is charity. Consequently, they work for the projects because they enjoy being part of the community. The exchange of “goods for money” is superseded by “gifts for reputation” [cf. Koot06, 50].

In 2002, the Boston Consulting Group generated a survey concerning the motivation working for Open Source Projects which are listed on SourceForge.org. In Illustration 3 the reasons are given [cf. Koot06, 50].

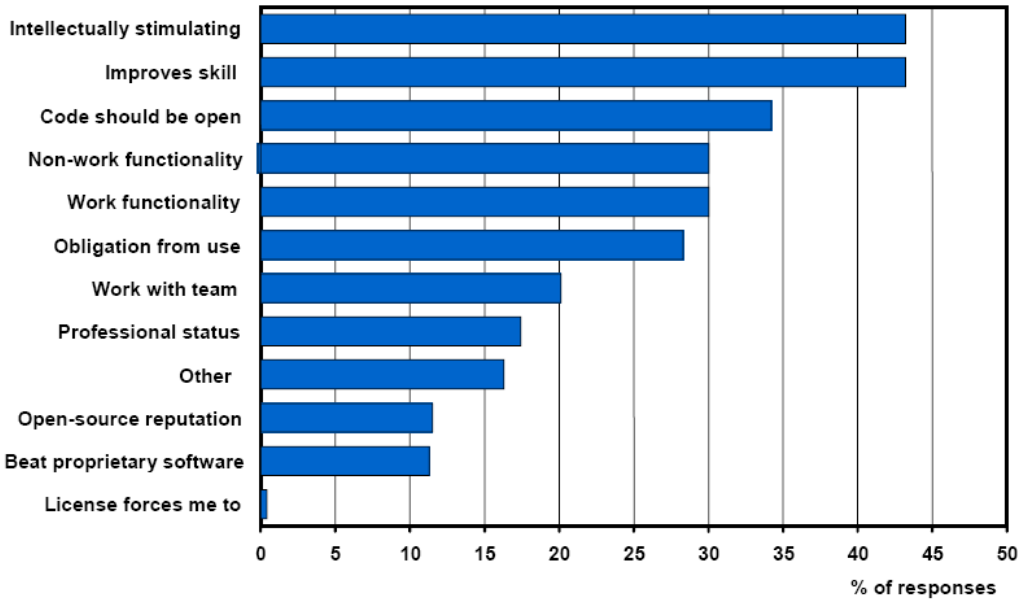


Illustration 3: Motives of Open Source Developers [Koot06, 51]

The main reasons why these developers work on Open Source Projects are “intellectually stimulating”, “work with team” or “nonwork functionality”. These arguments apply to other leisure activities, too. Furthermore ideological motives, like “code should be open”, can also be found in other leisure activities, too. Finally, Open Source Development can be viewed as a normal form of leisure activity [cf. Koot06, 50 et seq.].

However, working with Open Source Projects involves several motives like “improve skill”, “work functionality”, “professional status” and “Open Source Reputation” [cf. Koot06, 51].

Considered from the economic point of view developers will only participate in a project, if the activity is combined with a positive profit for that developer as compared with his costs. In this case, the expense for the developer is time which he has to invest in the programming activity. The extent of the costs will be dependent on the gratification of the work. The consideration can happen either directly or ex post [cf. Koot06, 51].

The direct consideration is carried out for a developer of proprietary software by the means of salary. In contrary, the directly problem solving is the consideration for a programmer of Open Source Software. In case a developer only makes small development contributions within an Open Source Project, these contributions are associated with little effort and low costs for a high qualified developer. The availability of the source code makes it pretty easy to solve some smaller problems. Besides, the opportunity costs of sharing the new code with other users are rather low. If, in addition, the new modifications are relatively uninspiring the protection of this innovation does not pay. Moreover, the costs for making the modification available to the public are low, and so the individual developer incurs only minimal costs in making a disclosing and development contribution [cf. Koot06, 51 et seq.].

Not only working on small developments is worth the trouble, but also the dealing with subjects for a longer period of time is lucrative and achievement-oriented. However, programming can be a part of an ongoing education and is therefore an investment in the developer’s future career. The prospect of an eventually better career is the consideration of the work done now [cf. Koot06, 52].

Another kind of consideration for developers is to show their programming skills (signaling) and to have these valuated. The modified product can be used in a secondary market (labor market) because of having sent out signals regarding the quality of the product. So the work is used for signalling and to build up a good reputation. It is mentionable that such reputation can be found in the scientific sector like publications which are printed in magazines in order to achieve a good reputation [cf. Koot06, 52].

Only in the case, if a considerable development contribution is made, developers can build up their reputation and can use it consequencly in other areas. The bigger the proprietary market is, in which the reputation can be gained, the higher the incentive will be to take part in an Open Source Project. There are several factors which influence the range of the signalling effect [cf. Koot06, 52 et seq.]:

- It can be mentioned that the signalling effect is strong, if the skills of a developer are evaluated by a large group of people. Consequently, larger projects are more interesting and attractive for developers than small projects whose further development is yet nebulous.

- In the same way the challenging programming tasks have a strong signalling effect if the developer group is able to assess and evalutate the development contribution. The main focus in this connexion is the so-called “peer review”. Qualified programmers have to report about the programmer’s work whether it is good or not and their evaluation is responsible for a certain reputation. So it illustrates the benchmark for the quality of a product [cf. Koot06, 53].

There are indicators which represent the importance of signalling the practical design of Open Source Projects:

- The stating of the development contribution plays an important role in projects. In the normal case all developers which are involved are listed in the project history. To state an example – you can find all registered projects and the developers with their qualifications and contributions on the web page of SourceForge.

- In the beginning of Open Source Projects the main aim was to find predominantely, technically challenging solutions for operating systems. These solutions are best suited for reputation and therefore they offer a higher signalling incentive than other projects.

- The allocations to the individual contributions are very easy due to the modular structure of many projects [cf. Koot06, 54].

To sum up it can be said that there are numerous economic explanations for participating in Open Source Projects for developers, as they can use their participation to improve their career prospects. Nevertheless, it has to be mentioned that a proprietary software market is indispensable for such economic incentives [cf. Koot06, 54 et seq.].

4 Bibliography

4.1 List of Literature

[Gras04] Volker Grassmuck: Freie Software – Zwischen Privat- und Gemeineigentum, Bundeszentrale für politische Bildung, Bonn 2004, also available online: http://freie-software.bpb.de/Grassmuck.pdf, retrieved on 26. June 2006.

[Horn01] Axel H. Horns: Anmerkungen zu begrifflichen Fragen des Softwareschutzes, in: GRUR 2001/1, page 1-16.

[Jaeg06] Jaeger, Till; Metzger, Axel: Neues Recht für freie Software, c't – Magazin für Computertechnik, 4/2006.

[Khar04] Kharitoniouk, Svetlana; Stewin, Patrick: Grundlagen und Erfahrungen – Einleitung, in: Gehring/Lutterbeck (Hrsg.), Open Source Jahrbuch 2004 – Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin 2004, page 1-15, also available online: http://www.opensourcejahrbuch.de/2004/OpenSourceJahrbuch2004.pdf, retrieved on 20. June 2006.

[Kogl04] Koglin, Olaf; Metzger, Axel: Urheber- und Lizenzrecht im Bereich von Open-Source-Software, in: Gehring/Lutterbeck (Hrsg.), Open Sourc Jahrbuch 2004 – Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin 2004, page 293-304.

[Matu03] Matuska, Martin: Kategorisierung von Open Source Projekten – Aufbau- und Ablauforganisation, Diploma Thesis, Vienna University of Economics and Business Administration 2003.

[Müll99] Müller, Martin: Open Source– kurz & gut: O'Reilly Verlag 1999.

[Widm06] Widmer, Ursula; Bähler, Konrad: Open-Source-Lizenzen – Wesentliche Punkte für Nutzer, Entwickler und Vertreiber, in: Gehring, Robert

(Hrsg.), Open Source Jahrbuch 2006 – Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin 2006, page 165-179,
also available online:
http://www.opensourcejahrbuch.de/2006/pdfs/OpenSourceJahrbuch2006_online.pdf, retrieved on 26. June 2006.

[Wieb04] Wiebe, Andreas: Internetrecht – Zivilrechtliche Rahmenbedingungen des elektronischen Geschäftsverkehrs, Springer-Verlag, Wien 2004.

4.2 List of Online Literature

[Boek06] Boeken, Arnd: Die Common-Public-License – v.1.0 (CPL):
http://www.anwaltskanzlei-cvw.de/deutsch/faelle/EDV-%20und%20IT-Recht/Common%20Public%20License%2010-04.pdf,
retrieved on 10. June 2006.

[Broc06] Brockhaus GmbH: Studie Open-Source Applikationsserver,
http://www.brockhaus-gruppe.de/files/Seminararbeit.pdf,
retrieved on 30. May 2006.

[Ever06] Evers, Steffen: An Introduction to Open Source Software Development,
http://user.cs.tu-berlin.de/~tron/opensource/opensource.pdf,
retrieved on 18. April 2006.

[Foku06] FOKUS - Fraunhofer Institut für Offene Kommunikationssysteme: Open-Source-Lizenzen: http://openfacts.berlios.de/index.phtml?title=Open-Source-Lizenzen; retrieved on 30. May 2006.

[Frau06] Fraunhofer Gesellschaft: Renner, Thomas; Vetter, Michael;Rex, Sascha; Kett, Holger: Open Source Software: Einsatzpotenziale und Wirtschaftlichkeit: http://www.e-business.iao.fraunhofer.de/docs/fhg_oss_studie.pdf,
retrieved on 4. June 2006.

[Gree06] Greene, Thomas: Linux is a cancer, http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer, retrieved on 18. April 2006.

[Hohe04] Hohensohn, Heidi; Bretschneider, Ulrich; Renk, Stefan: Open Source-Software-Release: Ein Ratgeber für die Veröffentlichung von Software unter dem Open Source-Status, Universität Paderborn 2004, http://www.c-lab.de/fileadmin/redactors/data/Services_Downloads/C-LAB_Reports/1_C-LAB-TR-2004-1-Open-Source-Software-Release.pdf, retrieved on 13. May 2006.

[ifro06] ifrOSS – Institut für Rechtsfragen der freien und Open Source Software: License-Center: http://www.ifross.de/ifross_html/lizenzcenter-en.html, retrieved on 4. June 2006.

[Koot06] Kooths, Stefan; Langenfurth, Markus; Kalwey, Nadine: Open-Source Software – An Economic Assessment: http://mice.uni-muenster.de/mers/mers4-OpenSource_en.pdf, retrieved on 4. June 2006.

[Land06] Landesgericht München: Urteil: http://jbb.de/urteil_lg_muenchen_gpl.pdf, retrieved on 30. May 2006.

[Lutt06] Lutterbeck, Bernd; Gehring, Robert; Horns, Axel H.: Sicherheit in der Informationstechnologie und Patentschutz für Softwareprodukte – ein Widerspruch?, http://ig.cs.tu-berlin.de/forschung/IPR/LutterbeckHornsGehring-KurzgutachtenSoftwarePatente-122000.pdf/file, retrieved on 18. April. 2006.

[Magi06] Magistrat der Stadt Wien, MA 14; Lutz, Brigitte; Potakowskyj, Johann; Richter, Klaus; Starnberger, Kurt; Weissenberger, Robin: Studie OSS. Open Source Software am Arbeitsplatz im Magistrat Wien: http://www.wien.gv.at/ma14/pdf/oss-studie-deutsch-langfassung.pdf, retrieved on 23. May 2006.

[Newm06] Newman, Nathan: From Microsoft Word to Microsoft World: How Microsoft is Building a Global Monopoly, http://netaction.org/msoft/world/MSWord2World.html, retrieved on 18. April 2006.

[Nütt06] Nüttgens, Markus; Tesei, Enrico: Open Source – Produktion, Organisation und Lizenzen: http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft157/heft157.pdf, retrieved on 30. May 2006.

[O.A.06a] O.A.: The Free Software Definition, http://www.gnu.org/philosophy/free-sw.html, retrieved on 18. April 2006.

[O.A.06b] O.A.: The GNU Manifesto, http://www.gnu.org/gnu/manifesto.html, retrieved on 18. April 2006.

[O.A.06c] O.A.: Overview of the GNU System, http://www.gnu.org/gnu/gnu-history.html, retrieved on 18. April 2006.

[O.A.06d] O.A.: Open Source Initiative OSI – OSI History, http://www.opensource.org/docs/history.php, retrieved on 19. April 2006.

[O.A.06e] O.A.: Open Source Initiative OSI - The Open Source Definition, http://www.opensource.org/docs/definition.php, retrieved on 19. April 2006.

[O.A.06f] O.A.: Shareware, http://de.wikipedia.org/wiki/Shareware, retrieved on 27. May 2006.

[O.A.06g] O.A.: Freeware, http://de.wikipedia.org/wiki/Freeware, retrieved on 27. May 2006.

[O.A.06h] O.A.: Was ist Freeware?, http://www.freewarepage.de/freeware.shtml, retrieved on 27. May 2006.

[O.A.06i] O.A.: Gemeinfreiheit, http://de.wikipedia.org/wiki/Public_Domain, retrieved on 27. May 2006.

[O.A.06j] O.A.: What is Copyleft?, http://www.gnu.org/Copyleft/Copyleft.html, retrieved on 20.May 2006.

[O.A.06k] O.A.: Die GPL und freie Software: http://www.oreilly.de/german/freebooks/gplger/pdf/001-024.pdf, retrieved on 30. May 2006.

[O.A.06l] O.A.: heise online – Deutsches Gericht bestätigt Wirksamkeit der GPL: http://www.heise.de/newsticker/meldung/49377, retrieved on 30. May 2006.

[O.A.06m] O.A.: License – Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/License_agreement, retrieved on 2. May 2005.

[O.A.06n] O.A.: Council Directive on the legal protection of computer programs (91/250/EEC): http://eur-lex.europa.eu/LexUriServ/site/en/consleg/1991/L/01991L0250-19931119-en.pdf, retrieved on 15. June 2006.

[O.A.06o] O.A.: Council Directive on unfair terms in consumer contracts (93/13/EEC): http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31993L0013:EN:HTML, retrieved on 15. June 2006.

[Quad06] Quade, Jürgen; Kunst, Eva-Katharina: Linux-Treiber entwickeln, http://ezs.kr.hsnr.de/TreiberBuch/html/sec.lizenz.html, retrieved on 30. May 2006.

[Renn06] Renner, Thomas; Vetter, Michael; Rex, Sascha; Kett, Holger: Open Source Software – Einsatzpotenziale und Wirtschaftlichkeit: http://www.e-business.iao.fraunhofer.de/docs/fhg_oss_studie.pdf, retrieved on 30. May 2006.

[Stall06a] Stallman, Richard: The GNU Project, http://www.gnu.org/gnu/thegnuproject.html, retrieved on 18. April 2006.

[Stall06b] Stallman, Richard: Linux and GNU, http://www.gnu.org/gnu/linux-and-gnu.html, retrieved on 18. April 2006.

4.3 List of Licenses

[Apache] Apache License, Version 2.0, http://opensource.org/licenses/apache2.0.php, retrieved on 30. May 2006.

[Artistic] The Artistic License, http://opensource.org/licenses/artistic-license.php, retrieved on 30. May 2006.

[BSD] The BSD License, http://opensource.org/licenses/bsd-license.php, retrieved on 30. May 2006.

[GPL] The GNU General Public License (GPL), http://www.opensource.org/licenses/gpl-license.php, retrieved on 30. May 2006.

[GPLv3] GPLv3 Draft, http://gplv3.fsf.org/draft, retrieved on 13. June 2006.

[LGPL] The Lesser General Public License (LGPL), http://opensource.org/licenses/lgpl-license.php, retrieved on 30. May 2006.

[MPL] Mozilla Public License 1.1, http://opensource.org/licenses/mozilla1.1.php, retrieved on 30. May 2006.