2006

Selection of an Open Source License: Aspects for consideration for Organisation in the choice of an Open Source License.

Darren Skidmore
Monash University, dcskidmore@gmail.com

Selection of an Open Source License: Aspects for consideration for Organisation in the choice of an Open Source License.

Darren Skidmore
Monash University
Caulfield School of Information Technology
darren.skidmore@infotech.monash.edu.au

Abstract

This paper is a contribution to the understanding and management of Open Source Software (OSS), in discussing the issues of license choice and how to match this to organisational aims, in the adoption and usage of Open Source Software. License choice governs the obligations of an organisation for their software. The paper describes which issues in an OSS license which are applicable to any OSS license, thereby being generic or de jure issues, and those which are outcome issues, allowing for the fulfilment of organisational objectives, either from the licensor or licensee perspective.

Keywords

Open Source Software, Libre / Free and Open Source Software, Business decision, OSS, Value, software license

INTRODUCTION

This paper is a contribution to the understanding and management of Open Source Software (OSS), in discussing the issues of software license choice and how to match this to the organisational aims, in the adoption and usage of Open Source Software. The purpose of looking at license choice is that it is the conditions of the license which govern how the software may be used, and the options and obligations imposed by the license conditions onto organisation in how they can either use the software or ensure others use the software. There is a growing use of OSS within organisations; indeed Gartner predicts that 90% of the Global 2000 organisations will have formal OSS acquisition and management strategies by 2010 (Driver and Weiss, 2005). Although there has been work explaining the terms and details of OSS licenses, (St. Laurent, 2004, Välimäki, 2005, Rosen, 2004, Fitzgerald, 2003) there has been little written on the reasoning of organisations in choosing a license, Dusollier et al. (2004) discusses choices for the European Union, Fogel (2005) has some discussion of choosing licenses when running an Open Source project, but this work is not aimed specifically at Organisations, also there has been analysis of two major OSS licenses and the Microsoft EULA (Edwards, 2005) and articles on the choices of licenses preferable for a developer of software (Michaelson, 2004). The focus of this paper in on the licenses for source code and software, and the options available for the business decision relating to that software. Remembering that the organisational aims for one software application maybe completely different from the aims of another software application. Although an important companion to the software there will only be a brief comment on the licenses pertaining to documentation and standards, which also have relevance in the considerations of the software, although in some respects they are independent, and would follow the decision about the software license, in the case of documentation, or can be completely separate, in terms of the use of standards.

Although software can be sold, in general it is licensed (von Krogh and von Hippel, 2003), and it is the license which gives the source code and application the legal attributes which create the obligations and governance usage of the software, wether the software is Open Source or Non Open Source. With regards to Open Source Software licenses, there are a vast number of licenses which claim to be open source, the ifrOSS site lists over 180 (ifrOSS, 2005), the Free Software Foundation almost 100 (Free Software Foundation, 2005) and a comparison of the aspects of 52 licenses has been made (Commonwealth of Massachusetts, 2004). The Open Source Initiative (OSI) also has a list of licenses, but there is no commentary on them nor taxonomy, just a list. The term Open Source can be a misleading one; in that there is no actual mandated definition of what is an Open Source License. The Open Source Initiative (OSI) does have a list of 10 conditions which it will use to certify that a particular license is compliant to the Open Source Definition (Open Source Initiative, 2004). However, it is only a certification mark[1], but any organisation may claim to have an Open Source License although the specific license may not comply with the OSI's definition. However, although caution should be used, the mindshare of Open Source movement, and the majority of software which is considered to be Open Source are

[1] The license can be submitted to the OSI who will review the license against the 10 point OSD and then give a certification that the license conforms to the OSD and is therefore an OSI compliant license.

using licenses which are OSI compliant, with a subset of these being considered to be 'Free'[2] licenses. The largest being the GNU GPL[3] version 2 (Free Software Foundation, 1991). A major repository of Open Source Software, SourceForge.net[4] , has an analysis tool, FLOSSmole[5], can be used to discover the prevalence of the licenses. SourceForge lists out of over 700,000 projects with the highest number of projects using three licenses, the GNU GPL v2 at 66.7%, the GNU LGPL at 10.4%, and the BSD License at 7%, with the next most popular license used being under 2.5%[6]. These are specific instances of Open Source licenses, although there are many instances of Open Source Licenses, they can be grouped into a limited number of types of Open Source licenses (Skidmore, 2007), or in an alternative Fitzgerald (2006). Although there are other styles and types of licenses, the three licenses (the GNU GPL, the GNU LGPL and the BSD) represent the not just the three most commonly used licenses, but also the three major different types of licenses, in that one is a reciprocal license, one is a linking license and finally the other is non reciprocal, there are other types of open source licenses such as obligation licenses but these are discussed later in the paper.

The GNU GPL, is a reciprocal license, which been described as a propagating license (New Zealand State Services Commission, 2006), in that it is a requirement of the license that any future software that uses the source code, or part of the source code must be licensed under the GNU GPL license, although this does not apply if the software is just used internally[7]. This also has the effect of ensuring that the GNU GPL is self reinforcing in terms of being a dominant license. The GNU LGPL, the Lesser GPL (Free Software Foundation, 1999), is similar to the GPL, but allows other programs to link to the software application and use the output from that program, since it was originally written for software library applications, without requiring the other software to be licenses under the GNU LGPL or the GNU GPL. This has implications especially when using software which is licensed under the GNU LGPL and is being used in a software stack. This avoids the possible implication or fear that each application in the stack would be required to be licensed under the GNU. The final major license is the BSD or BSD style licenses, these licenses state that the original source code is available under an Open Source license, but that any future source code that is based upon the original source code can be used and released under other license conditions. In the past attribution of all of the past authors was required, but this has been lessened in some BSD style licenses, such conditions are why they are sometimes classified as Academic style licenses (Rosen, 2004).

Other types of OSS licenses may require that the original licensor can use any subsequent improvements commercially, or that the software is restricted in the industries or purposes which can use the software. As well there are license types which make the source code available under two or more licenses, commonly referred to as Dual Licensing, but more correctly the software is available for some uses and conditions under one license, and can be made available under another set of uses and conditions under a different license. With a Dual license situation, the Licensor must have ownership (legally) of all the source code. For further analysis of the license types see Skidmore (2007).

Before turning to thinking about the needs of an organisation in choosing an OSS license, some further background is needed on the conditions of the licenses. Although there are only a few major types of licenses, there are scores of different specific licenses, each with their own individual wording. The issue of license proliferation is something the Open Source Community is now attempting to address (Open Source Initiative, 2005, Rosen, 2005). Proliferation of the large number of licenses has been caused, in part, by organisations wishing to insert certain conditions, change the wording of terms, or where they required copyright over the license text, but to a large extent changes in the jurisprudence of software engineering and licenses over time has created the need for different licenses. Particularly strong has been the changes in recognition of consumer warranty laws, patents, and jurisdictional issues (St. Laurent, 2004, Välimäki, 2005, Rosen, 2004, Fitzgerald, 2003). In a report on license proliferation for the OSI, the number of licenses was not seen as the main problem, but discusses solutions such as agreeing on a standard wording for specific issues, the addressing issues such as those regarding Patents, ensuring the internationalisation of the licenses, and addressing jurisdictional issues to address the major proliferation issues (Rosen, 2005).

---

[2] In the Open Source environment, leaving aside the Non OSI compliant licenses, there are two types. The 'Open' which comply to the OSI's definition of Open Source Software the Open Source Definition. There is also the 'Free' licenses, which are a subset of the 'Open' Licenses, the Free licenses are those which comply with the four freedoms definition of the Free Software Foundation at {Free Software Foundation, #265}. The GNU licenses GNU GPL and the GNU LGPL are Free licenses.

[3] GNU's Not Unix – General Public License, Version 2

[4] http://sourceforge.net/

[5] http://sourceforge.net/projects/ossmole and http://ossmole.sourceforge.net/

[6] Query tool at http://floss.syr.edu/OssMole/index.jsp Query run "SELECT count(*) as total_records, code FROM `project_licenses` group by code" combined with "SELECT distinct(code), description FROM rf_project_licenses" Run on July 2006 data, on 5th August 2006.

[7] However, this does not mean an organisation must re-release any changes they have made to the code. If the changes are only used internally to the Organisation, they do not have to be shared. Even if the changed application is distributed, the clause in the GNU GPL v2 requires that at the minimum the source code is available to those who receive the Binary, the source code does not necessarily have to be distributed to all, only those who receive the Binary. Most organisations however, find it easier to make the source available publicly.

If an organisation is going to consider Open Source licensing then it will probably look at OSS licenses for two polar reasons, the first being that they may want to use Open Source Software, or code from Open Source Software, the other would be if they wanted to release their own software, for which they own the Intellectual Property rights, as Open Source[8]. Most of the issues between these two opposites are similar, the difference being that of the perspective. When wanting to use OSS, the questions to be asked are what conditions are we prepared to accept, what obligations are we already under and what are we prepared to subsequently be required to do. However, if an organisation wishes to release software under an Open Source License, then the questions become more of what do we wish others to be obliged to do, and what control do we want over derivative works, and in what can be a separate issue, what obligations do we want from modifications and additions to our software. However in this paper, except where there is a need to draw attention to a difference, we will treat the issues pertaining to the releasing under an OSS license and the use of software under an OSS license as the same, but where the suffix question to the issue is that of the licensor, or licensee.

CONSIDERATIONS OF WHAT SHOULD BE IN A LICENSE

With the OSS licences, or for that matter any software license, there are two major areas or aspects for consideration. The first aspect being generic issues that should be considered by every license and the second aspect that of the specific outcome choices for a business or organisational aim and decision. Separate to these issues are also other aspects in the form and style of the construction of the license, particularly those pointed out by Rosen (2005) in his analysis of the license proliferation issue. The licenses should be written in legal terms but attempt to be simple, and consideration should be given if there will be a need to have the license translated into another language.

These issues are considerations that should be made, if the choice is available, although it is quite probable that there is either limited choice or that there is no choice in some cases. For example, where the only option is that the GNU GPL license is required to be taken, some of the options are not available. The list below is formed from the jurisprudence of 2006, looking back, in the case of licenses that were developed over a decade ago, when some issues were either non issues, or the licenses were written specifically thinking about United States Law, and therefore did not always address some issues that are required by other jurisdictions. Some licenses, particularly the GNU GPL and the BSD licenses, were written because they are the practical embodiment of the philosophies of their organisations and therefore framed in certain ways.

It is also important to mention, that most of these issues are theoretical, in that of all the Open Source Licenses only the GNU GPL has been actually tested in a court of law, and was upheld (District Court of Munich, 2004). So although, there is legal discussion and some case law on similar issues in other contexts, there has been little actual legal precedent to give guidance, even the District Court of Munich's decision did not comment or address issues such as consumer warranty or Patent infringement. The issues of consumer warranty protection to a great extent and issues to deal with Patents to a much lesser extent have not been tested in relation to software, in this particular sense, certainly in Open Source Licensing, and at this point in time are more FUD[9] than actualised risks (Skidmore, 2006). Of the two issues, Patent infringement claims are more likely, and potentially more fatal, but some large software vendors who participate in Open Source have individually stated and collectively created a commons of patents and promised protection for Open Source projects from Patent infringement claims (Open Invention Network, 2005).

Generic / De Jure aspects

The generic aspects which really should be de jure aspects in OSS licenses (see Table 1), in the more modern licenses are issues which are common to all licences, and are aimed at addressing the jurisprudence of software use and software engineering. For any license these are important issues, not because of the effect that these issues will have on the licensor or licensee in terms of obligations but purely because of good governance and issues in software engineering.

Consumer Warranty
Patent issues
Choice of Law
Choice of Forum / Jurisdiction (may also be an Outcome choice)
Definitions of Derivative work, and Distribution
License mixing

Table 1 Generic / De Jure issues

[8] Some of the business reasons behind why an organisation wishes to go down the Open Source route will be discussed later in the paper
[9] FUD – Fear, Uncertainly and Doubt The on-Line Hacker Jargon File (2003) FUD. IN Raymond, E. (Ed.) *The on-line hacker Jargon File.* 4.4.7 ed.

There are two issues in Consumer Warranty. One is the fear that a developer would be held liable for not just errors in their code, but for faults that are caused by their code for not performing a task, even in the event that the developer never intended nor certified that the code could or should do that task. The other issue, is that of individual national consumer protection regulations, in that if for example under Australian law, where if there is a term in a contract or license which disclaims warranty then the clause (and possibly the whole license is invalided). Although both of these have been flagged as issues, and license terms drafted to deal, more particularly with the latter issue of limiting consumer liability, there does not seem to be any jurisprudence on the success or failure of such litigation.

The issue of Patents in ICT is a vexed issue, since patents are a monopoly on the control of an idea thereby allowing the patent owner to place an injunction on software infringing the patented idea, combined with the international issues of patents to create a difficult situation for Software developers (Skidmore and Skelly, 2003). Certainly there has been litigation in the past for Patent infringement, in software. The larger vendors in software development, not only create large patent portfolios, but seem to be constantly battling for or against Patent claims. Licenses in particular need to be clear about how they are going to deal with contributions which may in contain patented ideas, and also deal with issue of what to do when a patent claim is brought against the project / application.

Choice of Law is more a legal term which indicates under what type of law a dispute will use, some examples of types of law are intellectual property, sale of goods, and contract law. Although not a large issue, the choice of law will allow the licensor to control to some extent what type of law will be used if a dispute goes to court. Some licenses specifically exclude issues such as explicitly declaring that the choice of law will not be that of the sale of goods (Rosen, 2004). Other reasons for this are that there is more certainly under some types of laws because there is more case law, or to ensure certain outcomes.

The choice of forum or jurisdiction sets out in which country or states laws will be used, so the license might state that the laws of the state of Victoria, in Australia, or state of Delaware in the United States will be the law which is used in determining the case. Some times this is a de jure issue that is just a well thought out process to give surety, in others it is a deliberate outcome to ensure that the licensor controls the choice of Forum.

The issue of definitions, especially of the terms “derivative work” and “distribution”, is important, certainly in a license which is required to be international. The reason behind this is that in some jurisdictions in the world, including Australia, there is no definition of derivative work in the copyright legislation (Fitzgerald, 2003), so for this reason, describing and defining the terms should be done inside of the license. There are also definitional issues about what is meant by distribution. There is a general assumption that distribution means outside of the organisation, but in large multitiered, or multinational organisations where does the boundary lie? For instance with the Australian Government, does this just apply to the federal government, does it include the state governments, and local municipalities (Fitzgerald and Suzor, 2005)? Another issue with distribution is the increasing use of hosting services and use of technologies such as web services where a business might use OSS to provide software that other business use, but does not release the software because the software never leaves the original organisation, and is therefore not distributed. Amongst a great many other issues this is what the new version of the GNU GPL is attempting to address.

A final issue to be addressed is the issue of being able to mix licenses. In the case of some licenses, this is reasonably easy as the license, such as the BSD style, allows the original source code’s license to be changed. However in the case of reciprocal licenses, this can be much harder; for example with the GNU GPL, because the conditions of the license require that any resulting source code which uses source code licensed under the GNU GPL, then the resultant work must be then licensed using the GNU GPL. The mixing of code from two or more reciprocal licenses may not be possible because the resulting source code probably can not be licensed under both licenses, depending upon the conditions of those licenses. The draft European Public license (European Community, 2005), addresses this issue by providing a list of licenses which can be used if licensed under the EUPL was mixed with software licensed under other licenses, thereby allowing the mixing of source code. Previously such issues were limited, but given the growing number of applications using Open Source licenses, this is an issue which needs to be more explicitly addressed. Some applications are licensed under several licenses, to enable different outcome to be achieved[10].

Outcome Issues

The outcome issues are those which have an effect on how or what must be done with the source code, or which might effect the decision of the software developer in choosing the license or in ensuring aims and objectives of the licensor of the software. Listed in Table 2, these issues generally are beyond the software engineering

[10] For instance Mozilla Firefox is licensed under three different licenses, because of the needs of various stakeholders and their requirements for including or working with Firefox.

choices, and de jure legal issues of dealing with Patents or definitional issues but have an affect on the business risks and opportunities of the licensees.

Specifying conditions of modifications / derivative software including, Reciprocal / non Reciprocal nature Specifying linking nature Ownership / obligations of modifications / additions to the original source code
Jurisdiction
Issues of Trademark
Specification of limiting or declaring other Technology linkages or Field of Endeavour.

Table 2 Outcome issues

The most important of the outcome issues is that of specifying what is required of modifications, additions and derivative software which uses all or part of the original source code. Effectively these issues are the ones which make the most difference to subsequent use of the source code. If reciprocal, then any subsequent code must follow the conditions of the original license, if non-reciprocal then of course the resultant source code can then be licensed under other conditions not controllable by the original programmer / owner. Reciprocal conditions, again using the GNU GPL as an exemplar will generally (but conceivably not always) insist on ensuring that any new code is available to others. Whereas with non reciprocal licenses, subsequent source code does not have to be mandated to be made available to others, including the original creator of the source code.

A license can also be created, if the licensor wishes to ensure that others will have to reciprocate the source code if they modify / add / use the source code, but may allow other applications to link or use the original application. As is the case with the GNU LGPL, the license which was written by the Free Software Foundation, to address the issue of linking to application libraries and to specifically ensure that parties knew that linking was permitted, the original name of the GNU Lesser GPL was the GNU Library GPL, the name changing in 1999 with Version 2.1 of the GNU LGPL (Free Software Foundation, 1999). Linking is becoming an important issue in Open Source Software, partially because some organisations prefer to use software licensed under the GNU LGPL compared to the GNU GPL because then they can use the GNU LGPL licensed software applications in their software stack, and get the benefits of having open source software, but still be able to build proprietary applications to link into other components in the software stack. The vendor or organisation is then protected from relying upon a third party either going out of business or increasing the licensing fees to unsustainable levels, but is still able to generate value from other components in the software stack and not be required to share them with others.

Finally a licensor might consider, if they wish, on ownership or obligations on subsequent contributors to the original source. Some licenses require that any additions are able to be used by the original licensor, with minimal royalties to the author, or others that any code be required to be tested for compatibility to the original source code. There may also be conditions which restrict the choices of distribution of modifications.

Although it is also a de jure choice, some licenses will specifically require a jurisdiction, in the case of the EUPL, the choice is, if the source licensor is the European Commission then litigation will occur in the European Court of Justice, otherwise the jurisdictional court shall be that of the licensor. This is a choice that may be made explicit if a licensor wishes to ensure that they will not have to enter a dispute in an unfriendly or foreign (and potentially more expensive) jurisdiction.

Some software licenses, especially those of particularly important or prominent software applications specifically protect the trademark of the original application. For instance the Apache Foundation, who maintain the Apache httpd web server source code, although allowing the use of the source code, prohibit the use of Apache foundation trademarks to be used in any derivative software (Apache Software Foundation, 2004).

Under the OSI's Open Source definition, an Open Source license cannot restrict the product, other software and must be technically neutral, it must also not discriminate against any group or field of endeavour. If a license does any of these then it cannot get OSI certification as an Open Source license. However if there is an organisational aim or philosophy or a need to ensure that such conditions are required then this needs to be stated into the license. The license will not be OSI certified, and may certainly restrict those willing and / or able to participate, but can be inserted if that is the business decision of the organisation.

LICENSE CHOICE

The license choice as stated in the beginning may be forced upon some, with others there may be a choice amongst options for software or source code, where the license of the source code is one of the options to be weighted up against the quality of the code, the fit to the organisational or functional need and the cost of not using that particular source, module or application.

Licensor

Where an organisation is the licensor, generally the choice seems to be that of requiring of reciprocally, and perhaps of trademark protection. Reciprocally will in general ensure that subsequent source code is available to all, this might further the organisational needs in that improvements can be added back into the organisations own source tree, or that in being forced to be open that more external developers may be attracted to working and participating in the ongoing development. Reciprocally may also ensure that a rival cannot fully take the code and gain strategic or operational advantage over the originator, although this might depend on the definition of distribution, derivative and requirements to hand back modifications.

The issue of others using and participating in the ongoing [public] development, may also be enhanced if linking is allowed, so that others are not intimidated in participation by the belief that they will have to share other source code of applications which link to the code. Similarly, adding conditions for the sharing back of code, may assist the licensor in keeping a strong and focuses ongoing application, which is acceptable to the Open Source Community.

If an organisation wishes to have an active community participating in the ongoing development of the source code, and wishes to maintain or at least keep abreast of the changes then the license is not the only issue that needs attention. There has to be infrastructure to support a repository, and commitment from the organisation for the ongoing development or maintenance of the source, this can be very active, or less active, but does need to be considered (Fogel, 2005). There might also be participation by various means in the Open Source Community to keep participation and reputation within the community strong. Again this depends on the business reasons for the creation of the Open Source code. The majority of software applications require maintainers, so the ongoing costs of Open Source may or may not be less than support of a proprietary application.

Licensee

The issues for the licensee are similar as to those of a licensor, from the reverse angle. Again, certainly if there is no choice, but to use software licensed under the GNU GPL, some, especially, software vendors are reluctant to use the source code or application software.

However in the case where an organisation is an end user of software, the choice of software that is licensed under a reciprocal license might be of benefit. Although it is possible that an end user organisation could sell or gain from code built for them, the use of reciprocal OSS ensures that they can access ongoing improvements, patches and additions, without being beholding to a specific vendor. Remembering that the ongoing maintenance and patching of software is a management issue for all software not just Open Source Software. It is also possible for an organisation if it participates in the community to have the software project consider and build in the organisational objectives and needs. There is actually a risk in not participating by forcing the organisation to cross develop to maintain compatibility (Edwards, 2005), but this is also an issue in other non OSS applications such as when organisations customise Enterprise Systems software.

Some Licensees have attempted to abuse the open source nature of code and use reciprocal licensed code, specifically licensed under the GNU GPL, and then not return the modifications into the community. An organisation called GPL Violations.org[11] is dedicated to tracking down and bringing this to the community's attention, and if necessary legally forcing the infringing organisations to share or stop using the source code.

NON SOFTWARE OPEN SOURCE LICENSES

There are licenses applicable to documentation, artwork and other related intellectual property which an organisation can use similarly to software, for which some of the arguments are valid, although some of course are not since they are specific to the software. Some good examples of these are the Creative Commons licenses (Creative Commons, 2005), and the Free Software Foundations Free Documentation License(Free Software Foundation, 2002).

CONCLUSION

In the end the choice of an Open Source Software license, must be a business decision that achieves or fits with the organisations aims or purposes, it may also be a philosophical one as well, such are the examples of the European Union's choices. As with any ICT, leverage can be obtained from the ICT if the organisation can manage and understand both the aims of the organisation and what it requires from the ICT. This paper has looked at what is important to included in a license, although, because of space issues, it has not gone into details about the suggestions of what specific licenses would fit best the conditions, also, since the list of appropriate licenses changes and due diligence also must be taken in the practical implementation of using an Open Source

[11] http://www.gpl-violations.org

License. Further work needs to be done on both the enactment of tightening up of the license terms as well as the issue which are seen as important to organisational adoption. However in conclusion, it is possible to leverage Open Source Software and to profitably and successfully use it in the organisation.

REFERENCES

Apache Software Foundation (2004) Apache License, Version 2.0. Apache Software Foundation, http://www.apache.org/licenses/.

Commonwealth of Massachusetts (2004) Open Source Licenses - Quick Reference Chart. IN Http://Www.Mass.Gov/Itd/Legal/Quickrefchart.Xls (Ed.), Commonwealth of Massachusetts.

Creative Commons (2005) Licenses Explained: Creative Commons. http://creativecommons.org/about/licenses/

District Court of Munich (2004) Harald Welte vs Deutschland GmbH. District Court of Munich.(unofficial English translation) http://www.jbb.de/judgment_dc_munich_gpl.pdf

Driver, M. & Weiss, G. J. (2005) Predicts 2006: The Effects of Open-Source Software on the IT Software Industry. IN Gartner (Ed.), Gartner.

Dusollier, S., Laurent, P. & Schmitz, P.-E. (2004) Open Source Licensing of software developed by the European Commission. IN Unisys (Ed.) *IDA/GPOSS Encouraging Good Practice in the use of Open Source Software in Public Administrations.* European Commission.

Edwards, K. (2005) An economic perspective on software licenses—open source, maintainers and user-developers. *Telematics and Informatics,* 22**,** 97-110.

European Community (2005) Draft-European Union Public Licence. 0.2 ed., European Commission. http://europa.eu.int/idabc/en/document/5425

Fitzgerald, B (2006) The Transformation of Open Source Software. MIS Quarterly, 30, 587-598.

Fitzgerald, B. (2003) *Legal Issues Relating to Free and Open Source Software,* Brisbane, Queensland, Australia, Queensland University of Technology School of Law.

Fitzgerald, B. & Suzor, N. (2005) Legal Issues for the Use of Free and Open Source Software in Government. *Melbourne University Law Review,* 29**,** 412-447.

Fogel, K. (2005) *Producing open source software : how to run a successful free software project,* Beijing ; Sebastopol, CA, O'Reilly.

Free Software Foundation (1991) GNU General Public License Version 2. 2 ed. http://www.gnu.org/copyleft/gpl.html#SEC1

Free Software Foundation (1999) GNU Lesser General Public License Version 2.1. 2.1 ed. http://www.gnu.org/licenses/lgpl.txt

Free Software Foundation (2002) GNU Free Documentation License (GFDL), Version 1.2. http://www.fsf.org/licensing/licenses/fdl.html#SEC1

Free Software Foundation (2005) Licenses. In Free Software Foundation (Ed.) *Licenses.* Free Software Foundation. http://www.fsf.org/licensing/licenses/

Ifross (2005) License Center. IN Software, I. F. R. D. F. U. O. S. (Ed.), Institute für rechtsfragen der freien und Open Source Software. http://www.ifross.de/ifross_html/lizenzcenter-en.html

Michaelson, J. (2004) There's no such thing as a Free (software) lunch. *ACM Queue,* 2.

New Zealand State Services Commission (2006) Guide to Legal Issues in Using Open Source Software v2. State Services Commission. http://www.e.govt.nz/policy/open-source/open-source-legal2/index.html

Open Invention Network (2005) Open Invention Network formed to promote linux and spur innovation globally through access to key patents. IN Open Invention Network (Ed.), Open Invention Network, http://www.openinventionnetwork.com/press.html.

Open Source Initiative (2004) The Open Source Definition. http://www.opensource.org/docs/definition_plain.php

Open Source Initiative (2005) License Proliferation. Open Source Initiative, http://opensource.org/docs/policy/licenseproliferation.php.

Rosen, L. (2004) *Open Source Licensing Software Freedom and Intellectual Property Law,* Upper Saddle River, NJ, Prentice Hall.

Rosen, L. (2005) License Proliferation. Open Source Developers Lab, http://www.rosenlaw.com/LicenseProliferation.pdf.

Skidmore, D. (2006) Too many Open Source Licenses! But do the existing licenses adequately encompass the diverse needs and concerns of particular stakeholders? IN Özel, B., Çilingir, C. B. & Erkan, K. (Eds.) *Towards Open Source Software Adoption: Educational, Public, Legal, and Usability Practices. OSS 2006 tOSSad workshop proceedings.* Como, Italy, TÜBİTAK (The Scientific & Technology Research Council of Turkey).

Skidmore, D (2007) *Forthcoming* FLOSS Legal, Engineering terms and a license taxonomy. IN St.Amant, Kirk & Still, Brian (Eds.) Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives. Idea Group.

Skidmore, D. & Skelly, L. (2003) Patents in Information Systems: International Issues. IN Ang, J. & Knight, S. (Eds.) *The 4th International We-B Conference.* Perth, Australia, We-BCentre, Edith Cowan University.

St. Laurent, A. M. (2004) *Understanding Open Source and Free Software Licensing*, O'Reilly.

The on-Line Hacker Jargon File (2003) FUD. IN Raymond, E. (Ed.) *The on-line hacker Jargon File.* 4.4.7 ed, http://www.catb.org/~esr/jargon/html/F/FUD.html

Välimäki, M. (2005) *The Rise of Open Source Licensing - A Challenge to the Use of Intellectual Property in the Software Industry*, Turre Publishing.

Von Krogh, G. & Von Hippel, E. (2003) Special issue on open source software development. *Research Policy,* 32**,** 1149-1157.

COPYRIGHT