Open Source Software: Risk Management from an Intellectual Property Perspective

Kanav Hasija[1], Badri Narayanan[2]

[1]Dept Mechanical Engineering, Indian Institute of Technology, Kharagpur, India, kanav.hasija@iitkgp.ac.in

[2] IP Lawyer, Lakshimkumaran & Sridharan, New Delhi, India, badri@lakshmisri.com

Abstract

Increasing efforts are being made to 'outsource' research, build innovation networks, and collaborate with research institutions for delivering services and goods to clients and customers. Open Source allows companies to reduce their research and development time and cost and receive innovative solutions developed by experts in their fields. However, in this world of bold CEOs and bolder thought leaders, Open Source is treated with concern and trepidation. These concerns arise due to the perceived legal and technical risks associated with Open Source.

An approach to utilize Open Source must start with the understanding of the legal and technical risks associated with Open Source. Typically, the users do not comprehend the ambit of their legal rights under the license, the liability entailed due to the breach and the effect of such breach on their clients and customers. On the technical front, the main concerns relate to issues of compatibility, reliability and security. Another major deterrent is the lack of effective technical support.

Organizations can mitigate the legal risks in a variety of ways. There are simple step methods and complicated processes to achieve this. A simple method of mitigating the risk is by having an inventory management system of the actual and potential open source software that the organization would like to use. Inventory management system can give clear snapshot to a company about maturity of software and how useful will it be for the company.

1. Introduction

In this rapidly advancing world, companies have to leverage intellectual power outside their organizations to sustain growth and competition. Sustaining the growth and stability of their business, everyone is trying to protect their intellectual property and fight against their competitors for the same. Open source industry is growing a lot because of their innovative business model. A 2004 InformationWeek survey found that 67% of companies use OSS/FS products.Open Source community has top level programmers, intellectually sound volunteers, debuggers and what not. This sharing-basis model has given much good software for the past years.

The only concern people fear from using the open source software is no guarantee and indemnity protection from intellectual property infringement. Anyone can contribute to any open source project after a sound quality check is been carried out for the segment of the program contributed. There may be cases when someone has contributed to the project from someone’s intellectual property right, and doing so may cause contamination of other’s intellectual property in open source software code which is not been kept on the check list.

This paper helps one to understand the risk involved in open source software and talks about taking care of the maturity of the open source software, for running the business in profits, and intellectual property of others to be on a safer side from heavy law suits. Inventory management of both the maturity of code and infringement on other’s intellectual property is the best and safest deal to play before incorporating or developing open source software.

2. Intellectual Property and Software

There are certain group of rights enjoyed by the inventor of his/her intellectual work, if he/she is protected under them. There are various kinds of intellectual property rights like patents for protecting products, copyright for protecting artistic work, trademark for protecting distinguishing symbols and much more. Software can be protected only by two fields of intellectual property rights namely patents and copyrights.

2.1 Patent

A patent is a set of exclusive rights granted by a state to an inventor or his assignee for a fixed period of time in exchange for a disclosure of an invention. After the patent has been granted by the state, no one can violate the inventor’s rights without his/her consent. If it happens so, knowingly or unknowingly infringer, who has infringed inventor’s rights has to bear the compensation for the damages caused to the inventor in said act. EMC and Hewlett Packard went in the court for patent infringement over StorageApps and HP had to settle the case by paying EMC a lump sum of $350 million.

2.2 Copyright

Copyright is a legal concept, enacted by governments, giving the creator of an original work of authorship exclusive rights to it, usually for a limited time, after which the work enters the public domain. Generally, it is "the right to copy", but also gives the copyright holder the right to be credited for the work, to determine who may adapt the work to other forms, who may perform the work, who may financially benefit from it, and other, related rights.

3. Open Source Software

The term open source software generally refers to software that is made available with its source code inclusive of the right to modify the software and the right to distribute the modified work. The basic difference between open source software and all other software is that source code of the software is available for free to use, modify or distribute which makes it far more flexible than most of the proprietary software. Though there is one misconception between free and open source software, that both are meant to be the same. This mystification will be cleared in section 3.1.

3.1 History

In the early days of computing (approximately 1945 to 1975), computer programs were often shared among developers, just as OSS/FS practitioners do now. An important development to OSS/FS was the start of the ARPAnet, the early form of the Internet. Another critical development was the operating system Unix, developed by AT&T researchers, and distributed as source code with modification rights for a nominal fee. However, as years progressed, and especially in the 1970s and 1980s, software developers increasingly closed off their software source code from users including Unix. Richard Stallman, a researcher at the MIT Artificial Intelligence Lab, found this closing of software source code intolerable. In 1984 he started the GNU project to develop a complete Unix-like operating system which would be Free Software (free as in freedom, not as in price, as described above). In 1985, Stallman established the Free Software Foundation (FSF) to work to preserve, protect and promote Free Software; the FSF then became the primary organizational sponsor of the GNU Project. The kernel architecture of Unix was on a start to be re-designed and the improved system now available in the market known as Linux/GNU operating system is an output of this initiative.

In spring of 1997, a group of leaders in the Free Software community gathered, including Eric Raymond, Tim O'Reilly, and Larry Wall. They were concerned that the term "Free Software" was too confusing and unhelpful (for example, many incorrectly thought that the issue was having no cost). The group coined the term "open source" as an alternative term, and Bruce Perens developed the initial version of the "open source definition" to define the term. Immediately after this, in 1998 Open Source Initiative (OSI) was formed to control and unify the definition of open source and certifies open source licenses associated with open source software.

4. Open Source Licenses

Open source software is copyrighted intellectual property which is licensed under a generally agreed upon set of principles, which are formally described by the Open Source Initiative at www.OpenSource.org. Open Source licenses are generally very flexible as compared to tough rules set by other software licenses. Some of the important features of the rules are: free redistribution, free availability of source code, allowing distribution of derived works and complying with the open source software license on further distribution.

Availability of source code for free attracts many users without falling into the depth of all license terms. Supposedly, a user derives new code from the open source software's code and distributes the software without supplying the source code for free or he/she prepares a new license agreement which doesn't comply with the original license, then the user is liable to face the court for a breach of license agreement. In yet another case, if a proprietary software developer has developed a code containing an open source code, which is not in his knowledge, he might end up in a breach of license agreement because he didn't supply the source code of the open source element along with the software.

Thus one can say that, handling open source licenses is an important task while implementing open source software in their models. It is yet another important task to choose an appropriate open source license for the original author while he/she distributes his/her software for the very first time. Given below in Table1 are some popular open source licenses and their characteristics.

Table 1. Some popular open source licenses

Name	**Characteristics**
GPL License	The ubiquitous open-source license. Requires all contributed code to be returned to the community.
LPGL License	GPL and open source software can be binary linked to proprietary programs in certain circumstances.
Artistic License	Copyright holder maintains some appearance of artistic control over the development of the package, while giving the users of the software enjoy the right to use and distribute the software in a more-or-less habitual fashion.
BSD License	Includes legal disclaimer of liability with explicitly named organization.
Apache License	Requires derivative works to provide notification of any licensed or proprietary code in a common location.
Mozilla Public License	Allows liberal mixing with proprietary software.

5. Risks Involved

Apart from breach of license agreement, there are various other factors associated with using the open source software. These risks have been divided into two sections namely legal and technical risks for better understanding.

5.1 Legal Risks

As discussed earlier in section 2, there is a high possibility of contamination of someone else's intellectual property in an open source project as there many contributors to the same project. Most secured way to mitigate these risks is prior art search on all the codes used in the project to check if someone already has an intellectual property on the code and if it has a worth. If any such intellectual property is tracked, one must find out a way to tweak around the prior art and safeguard himself from IP infringement. Another important legal risk is the breach of license agreement which must be dealt with utmost care and all the license terms must be complied with.

5.2 Technical Risks

According to the BCG Survey in July, 2002, 30% Open Source Software developers work as a part of their employment, over 50% are part of technical organizations and 20% are students. The quality and maturity of the code may get affected with diversity in the background of the developers. Thus, one of the immediate technical risks is quality of the code including factors like security, compatibility and testing. Another common risk seen in the open source software is lack of documentation and support for understanding the code better. This leads to fear of using or modifying the code as it is not easily understandable by a large part of the user community. These risks can be mitigated by choosing the software developed by popular developers in the community and maturity assessment of the code as mentioned in section 7.

6. Inventory Management

It is hard to track the contamination of proprietary software on IPR protected software in the open source project with open source community having more than 2,50,000 users and more than 50,000 developers registered on SourceForge. Open Source Risk Management (OSRM) warned Linux of infringing 283 patents of other parties including 25 patents of their biggest competitor, Microsoft. In this era of competitiveness, proprietary software developers are giving guarantees and indemnity protection to the users. It is essential for open source software developers to do the same for staying in the race.

Inventory Management is important to track all the risks involved in open source software and mitigate those risks. It gives a clear print of all dangers at once. Companies and project developers deal with huge number of lines of code, which makes it difficult for them to track each and every code all the time. The best possible solution in such a case is to prepare software architecture. An external inventory containing all possible codes one is using, which is of importance and has some value, is the first step to software architecture. Each code must be analyzed with the risks involved along with the maturity assessment.

The important factors to be taken in consideration while inventory management is:

- **License Agreement**: License associated with the open source software will aware you of the risks involved in case of breach of license agreements.
- **Intellectual Property**: If there is some intellectual property of other associated with the code you incorporated in the project, it may risk the infringement damages owner of intellectual property may claim at later stage. Thus, one is advised to perform a prior art search before incorporating a code in its project. Prior Art search is an exhaustive search on intellectual property which may be covering your code and is carried out by IP Consultants or law firms involved in intellectual property rights.
- **Maturity Assessment**

7. Open Source Maturity Assessment

Open Source Maturity Assessment (OSMA) is a systematic way to approach on a view of maturity of the code. The code may be very efficient in functioning, but certain aspects make it fit to be incorporated in professional software like security, quality, developer and documentation and support. Organization must prepare a list of possible factors affecting the maturity of the code and give weight to each factor for calculating a cumulative maturity score. According to the benchmarks set for the type of usage of code one can reach to a conclusion whether to incorporate a code in his/her project or not. Given below in table2 is a sample of factors and weights. Table3 gives a sample benchmark score according to the usage of the code.

Table2. Factors affecting maturity of code and their weights

Factor	Weight
Software Functionality, Quality and longevity of project, Extensibility, Development team, Security	4
Support From Community and developer	3
Training	1
Documentation Developer's releases, Web posting by others	2
Integration with various platforms	3

Table 3.Benchmark Score

Usage	Benchmark Score
Internal (Study 'n' Testing)	40%
Internal (Use)	50%.
Development 'n' Production	65%

8. Conclusion

Open Source Software is a big buzz in today's world because of its unique work-sharing model and savings in the R&D budget of many companies. Though the revenue generated from open source software is low, but the large user base compensates the loss. There are still companies fearing from using open source software because of no indemnity protection and guarantee. There are still some projects which don't go to completion and die in between of their progress. If open source developers start mitigating the risks which companies fear from, they have a long way to go for dominating open source software over proprietary software.

9. References

ActiveState 2008, *License to Code: Indemnifying Your Business Against Open Source Licensing Liabilities,* viewed 10 July 2008, <http://www.activestate.com/business_solutions/license_to_code.mhtml>

Bernard Golden 2004, *Succeeding with Open Source,* Addison Wesley Professional, San Carlos, CA

David A. Wheeler 2000, *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers,* viewed 10 July 2008, <http://www.dwheeler.com/oss_fs_why.html>

Olliance Group 2008, *Open Source Software Intellectual Property Management: An Introduction and Overview of Industry Best Practices,* viewed 10 July 2008, <http://www.olliancegroup.com/opensource/IP%20Management%20Best%20Practices%20in%20Open%20Source%20-%20Olliance%20Group.pdf>

Open Source Initiative 2008, Open Source Initiative, California, USA, viewed 15 August, 2008, <http://www.opensource.org>