Risk Model for OSS: Valuing hidden costs

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

IT based industries have shown great interest in Open Source Software (OSS) over the past few years. The fundamental reason for the exponentially expanded share of the OSS products in the industry may seem to be reduced R&D costs and time. The visible actual costs associated with OSS at first blush points to licensing costs. What people don't recognize while using the OSS are the hidden costs, which are realized sooner or later. An old joke on software goes like: Software, free. But hey the manual costs only $10,000. A severe threat to the open source industry is from the communities of patent holders who can at any point of time smack the open source licensees by filing an infringement suit.

It is valuable to understand the virtual costs and prepare a model to evaluate the risks and costs involved in using the OSS. We have tried to prepare a risk model for the same in an organized fashion. Without a single well-defined risk assessment technique and model, every company is using its own models to assess the risk (if it does so in the first place). Many ways can be implemented for leveraging the quality of the model, one of which has been discussed over here. A way of sharing modules with your external associates (R&D partner) can increase model's accuracy and reliability by choosing a common denominator to combine both the assessments.

1. Introduction

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. Zona research study found that over half of the large enterprise respondents expected increases of up to 25% in the number of GNU/Linux users in their firm, while nearly 20% expected increases of over 50%. Many companies are shifting a trend towards using open source software because of low cost of ownership. There are many failure projects found while using open source because many of them avoid looking at the hidden costs associated with the open source.

This paper tries to discover all the possible hidden costs associated with the open source and ways to track and mitigate those. Applying the risk model in your business as explained later will help firms implementing open source efficiently and cheaply.

2. 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. Free software and open source software were synonyms before the term open source was coined in 1997. This term was coined to clear the perplexity that all those software which gives their source code for free are not free, and all those software which are sold for free are not supplied with their source code. This is the basis of difference between open source software and free software. It also clarifies that free software developer do not want to share his intellectual property but open source developer love sharing their work so that more improvements can be done by a large programming open source community available for voluntary work.

3. 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.

One of the court cases in District Court of Munich clearly specifies the importance of license agreements in open source software. In *Harald Welte v. Sitecom,* No. 21 O 6123/04 , District Court of Munich, the plaintiff was the maintainer of the open source project ‘netfilter/iptables’. The defendant was a Dutch company which used the plaintiff’s software in their wireless network product without abiding by the terms of GPL license which was covering the plaintiff’s software. The claim for preliminary injunction by the plaintiff was upheld by the court and compensation by the defendant for the court hearing was delivered.

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 Table 1 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.

4. 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.

4.1 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.

4.2 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.

5. Hidden Cost

Open Source software has many advantages over proprietary software like: Flexibility and free availability of source code, reliable and large user base. There are many risks associated with it, which might lead to increase in cost of ownership. We shall discuss about the technical and legal risks separately for better understanding.

5.1 Legal Risks

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. Another risk involved in sharing your work in open source is dilution of your own intellectual property which might be valuable somewhere else.

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 6.

6. Risk Assessment Model

In-depth analysis of all the risks involved with open source explained in earlier section gives a fair idea about the risk assessment model an organization must follow before incorporating open source in their projects. Figure 1 explains one such model for risk assessment. Each element of the model is explained in further sub sections.

6.1 Search

The search department of the risk assessment model is responsible of searching all the possible open source software their organization might be interested to incorporate, use or use for modification for their software development. SourceForge is the biggest open source portal available on net from where complete information regarding the open source software or project may be obtained. After the search, complete information of all the results must be sent to the inventory department.

6.2 Inventory

The inventory department will prepare a list of all the software found out by search team along with their properties which might be useful in the risk assessment. Some of these properties are described below:

- Software name and version
- Project team
- Launch date of software
- Number of users and developers till date
- Activity of the forum
- Support and Documentation
- Compatibility with various platforms
- Flexibility of code

6.3 Quality Assurance

Quality Assurance team will analyze the quality, flexibility and usability of the software. There may be a case that the open source software doesn’t have good documentation and support and software developers at your organization are not capable to understand the open source software code. The service cost to modify the code by getting paid manuals, documentation and training from the open source project developers might add to the ownership of the software. The security, flexibility and usability of the software are also important factors for deciding upon the cost of ownership and the cost of ownership might go high in case of these factors are found missing. After analyzing all the points they will come on an exact figure of the cost of service and maintenance of the open source software.

6.4 Licensing Department

Licensing department will read the terms of the license agreements associated with the open source software. After reading all the terms, they will evaluate the cost of complying with license agreement that the organization would bear and the cost which the organization might have to give to the open source project owners in case of not complying with the license agreements.

6.5 Infringement Analysis

Infringement analysis department would do an exhaustive prior art search for finding any intellectual property which might be covering the software code. An infringement analysis test would tell if those shortlisted intellectual properties are really covering the code and pose threat.

6.6 IP Valuation and Damages

This department will value any intellectual property given to them and find out damages one has to bear in case of infringement of other’s intellectual property. It would give an estimate of cost the organization will have to bear in case of infringement. Further, it would give an estimate of the add-in intellectual property asset the open source software can give to the organization.

6.7 Profit Analysis

Profit analysis department will finally collect data of all the cost involved and revenue that software can generate. After collecting data it would come to a conclusion on whether to incorporate the open source software in the project or not.

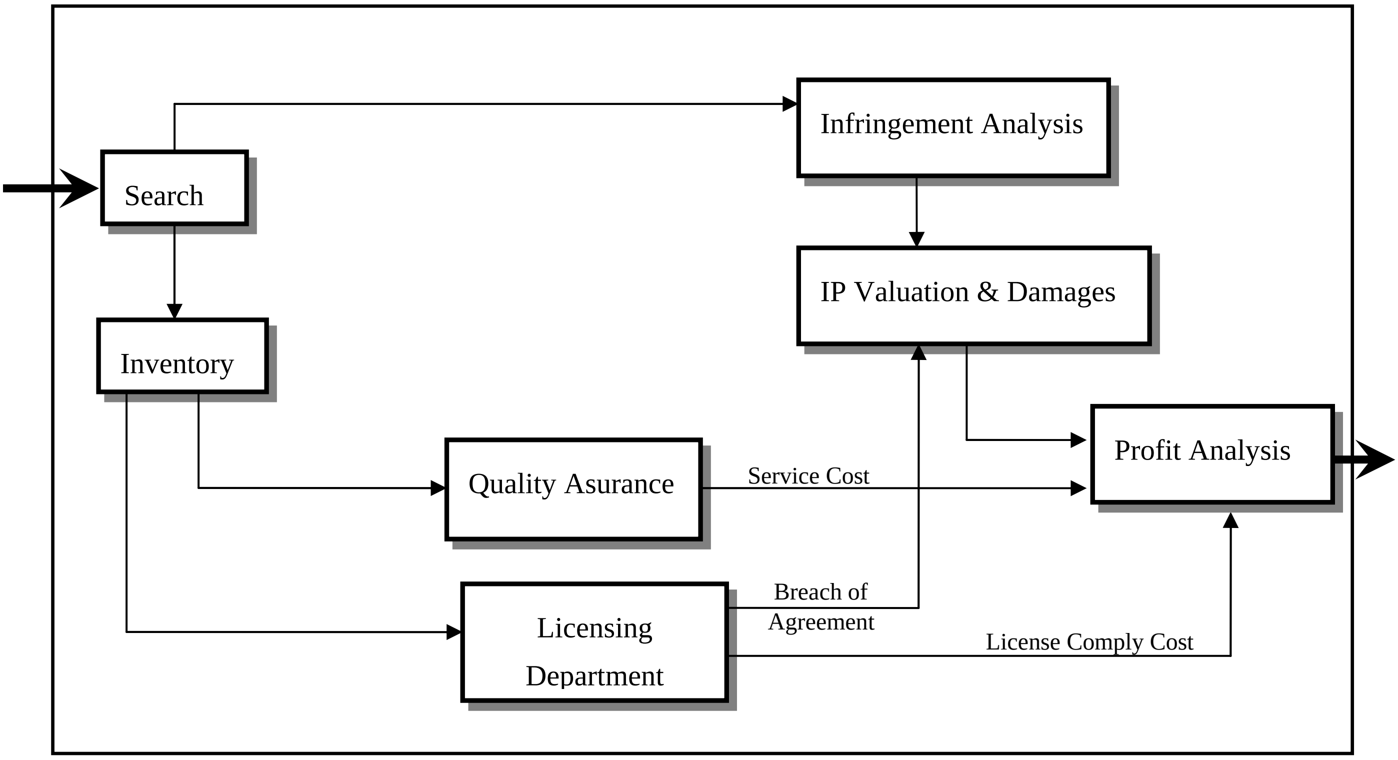

Figure 1. Risk Model (stand alone)

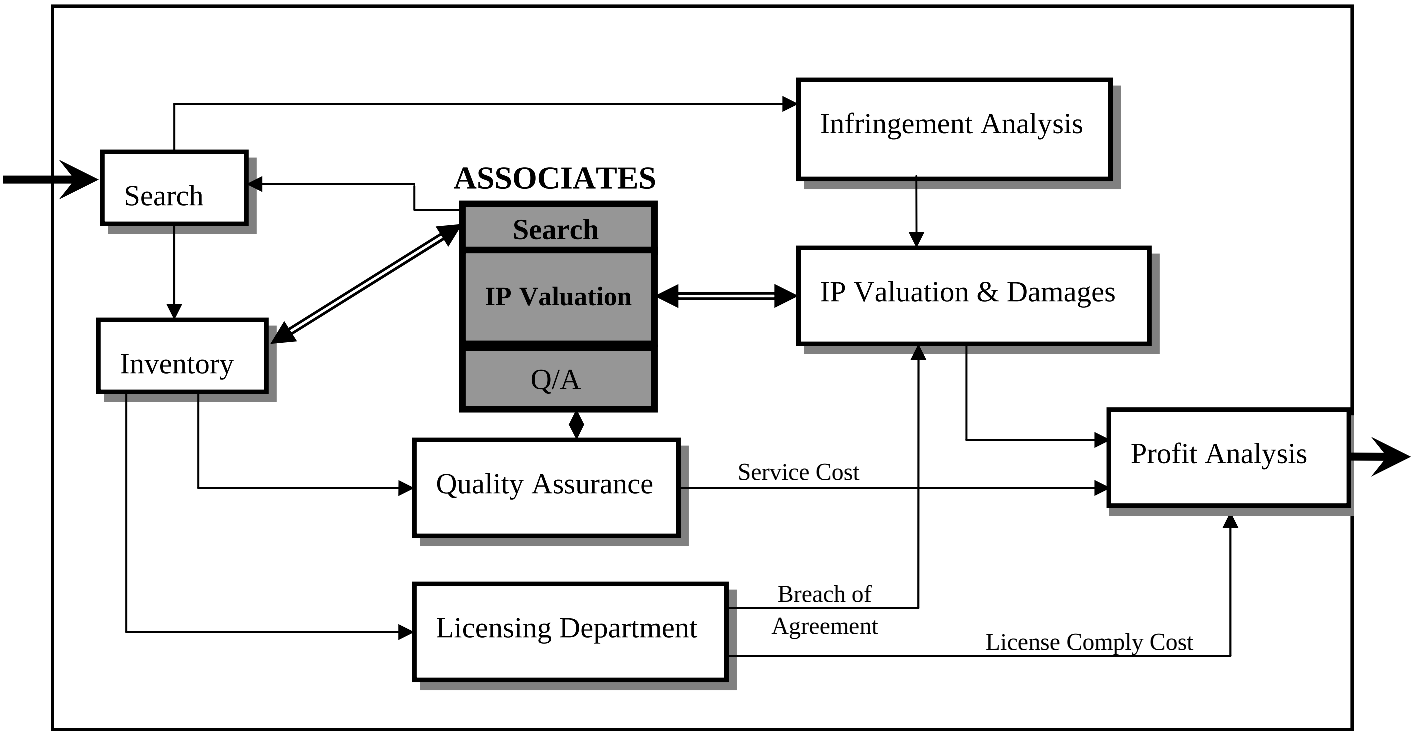

Figure 2. Risk Model embedding associates

It can further be noted that sharing the data, which has to collected by various departments, with the associate firms can the ease the risk assessment project and give a better efficiency and accurate results.

7. Conclusion

Open source phenomenon is helping today's industries by saving their R&D time and expenses. Considered to be relatively cheap, everyone is going after open source software without going deep into the hidden costs associated with the open source software. An in-depth analysis of hidden costs and risk involved in open source will benefit organizations to incorporate open source in their projects and benefit from them. A risk assessment model having search and data sharing with associates will guide organizations to understand the cost of ownership of open source software.

8. 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>