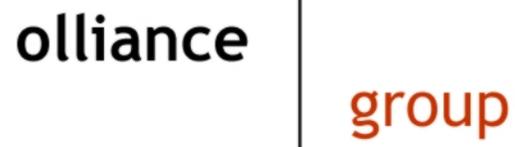


January 24, 2008



Open Source Software Intellectual Property Management

An Introduction and Overview of Industry Best Practices

Authored by Greg Olson
IP Management and Compliance Practice Manager
Olliance Group, LLC
www.olliancegroup.com

Foreword by Mark Radcliffe, Senior Partner, DLA Piper

Forward

This White Paper provides a cogent overview of the need to manage open source software as well as how to develop such management tools. The use of open source software has become ubiquitous in companies from the banks to hospitals. Open source software provides many advantages to its users, but its use requires careful management. Most companies use open source software in a hybrid environment, combining proprietary software with open source software. The failure to manage the use of open source software can result in serious problems: for example, the combination of software licensed under the General Public License with proprietary software can require that the source code of the proprietary software be made available without charge and with permission to modify and distribute without charge.

The use of open source software is carefully scrutinized both by investors and potential acquirers. In fact, the issue has become so important that several large companies that regularly acquire other companies have established a separate due diligence process focused solely on the use of open source software.

I have worked with Andrew Aitken and the Olliance Group on many client engagements and they remain the most knowledgeable consultants in the open source industry. I hope that you find this White Paper as useful as I did.

Mark Radcliffe
Senior Partner, DLA Piper

Table of Contents

Forward..... 2

Table of Contents..... 3

Introduction 4

What is open source software? 4

Why open source software requires special management..... 5

Discovery of license violations in redistributed code..... 7

Operational problems due to untracked open source software..... 7

Problems introduced through outsourced and contracted development..... 7

Problems with corporate acquisitions..... 7

Example..... 8

The benefits of managing open source software well 9

Fundamentals of open source software management..... 9

Know what open source software is in use and its essential attributes..... 10

Understand where each open source component is used..... 10

Identify who is responsible for the maintenance of each component..... 10

Complying with open source licenses 10

Best practices for managing open source software..... 10

Assessment 10

Strategy..... 11

Policy 11

Process..... 11

Repository 11

Success factors for open source IP management 12

Development ownership..... 12

Training..... 12

Tools for information sharing 12

Summary and Conclusion..... 12

About Olliance Group..... 14

Open Source Software Intellectual Property Management

An Introduction and Overview of Industry Best Practices

By Greg Olson
IP Management and Compliance Practice Manager
Olliance Group, LLC.

Introduction

This white paper will address the foundational components of successful open source software IP management, including challenges, success factors and best practices.

Open source software IP management is relevant to almost every type of software development today from infrastructure to applications, from large enterprises to small startups. Whether you are deploying open source solutions internally, building proprietary applications on an open source code base, planning for M&A or IPO, or even releasing a proprietary application as open source, you must be aware of the many facets involved in managing open source effectively, efficiently, and safely.

There are three primary elements of open source IP management. The first is the core strategy behind the use and management of open source software as a means to your business, technology and company objectives. The second focuses on open source code, licenses, and contracts. The third involves employment of best practices to manage potential risks and liabilities efficiently.

What is open source software?

The term open source software generally means 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. Open source grew from early roots in the academic world where software was developed through public or institutional funding and was freely shared by mandate. Open source software became a grass-roots software developers' movement in the 1980's fueled by frustration with the limitations of working with proprietary software systems that did not supply source code.

Open source software may be developed by a company, independent individuals or by public 'communities.' Communities are formal or informal gatherings of developers and other interested for the purpose of defining and refining software that solves a common need. The bodies of code that address a particular need are called "projects." Community-based projects are usually hosted on the Internet with tools that support their development community such as a forum and a public source code repository – a facility called a "forge." The code from these projects is freely and publicly available for developers to take, modify and embed into their own products/services.

Although open source software offers certain freedoms to software developers and users, it is not software in the public domain, nor is it necessarily free of cost. 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.

It is important to note, however, that we are aware of more than 400 licenses for source code available over the Internet, but only 65 of these have been reviewed and approved by the Open Source Initiative.

Users of open source software enjoy many rights that are granted through open source licenses. For instance, open source software licenses are generally characterized by the public availability of source code versions, and the right of any person or organization to modify and distribute such versions and any modifications they create. In this way, open source effectively invites anyone to participate in the growth and utilization of an open source project. But in order to maintain the integrity of the open source development model, open source licenses typically require certain reciprocal obligations on the part of the user and the distributor. Without these reciprocal obligations, such as public disclosure and right of access to source code, many of the advantages of the open source development model would be lost through redistribution. In order to understand the true impact of the reciprocal open source license terms, it is necessary to understand the way that open source uses the traditional notions of copyright, licensing, distribution and use.

The obligations of public source code availability and the perpetual integration of the license with the software distribution present the most serious implications for some users and integrators of open source software. The reciprocal obligations that arise from certain open source licenses such as the Gnu General Public License (GPL), for example, deserve the most attention and care. Users that combine their own software with software licensed under the GPL and distribute it are obligated by that license to make the source code of their combined code publicly available. Licenses that include such obligations are often referred to as “viral.”

Questions frequently arise about the manner of combination and scope of modification to open source software as it impacts license stipulations; this is referred to as “derivative work” and is often a condition that triggers terms of a license. Determining what is a derivative work is often quite difficult, and there exists little case law upon which to base decisions with confidence. Not only do the specific licenses have significant differences among them, but many, including the GPL, have gray areas that lead to different interpretations by both licensors and licensees. As with all manner of software licenses, it is wise to seek qualified legal counsel when basing business activity on the details of any license.

Although the licensing situation for open source software may seem messy and complicated, it is, in fact, far simpler than the range and variation seen in proprietary commercial software licenses, which number in the hundreds of thousands. The important point is that users and re-distributors of open source software, like users and redistributors of proprietary commercial software, must be aware of their rights and obligations under their licenses and take reasonable steps to insure their compliance.

Why open source software requires special management

Our extensive research has shown that companies with successful experience in using open source software have found that it requires special management for three reasons:

1. It has become nearly ubiquitous and easily seeps into many projects
2. It typically does not come with characteristics we have come to expect from commercial software
3. It is licensed software with specific rights and obligations that must be tracked

Open source licensing was conceived to promote the propagation and re-use of software, and in this regard it has been extremely successful. With the prominence of Linux, Apache, JBoss, Eclipse, MySQL and over 100,000 other active projects, proven open source implementations of innumerable functions are freely available on the Internet. Engineers who have been told throughout their whole careers “don’t reinvent the wheel,” naturally gravitate to these open source implementations as a way to speed their development projects. In many areas the forefront of innovation is occurring in open source communities, so open source is the best choice if one needs those leading edge features. In cases where a project requires interoperability with open source platforms such as Linux, Apache or Eclipse, use of open source code is unavoidable. Particularly among younger programmers the prevailing attitude about open source is “Everybody is using it and it’s really good, so it’s really stupid not to take advantage.” The fact that open source for just about any function is easily found on the Internet through on-line search services and that in most cases it is free to download immediately makes it nearly impossible to resist.

It is important to understand that these inducements to use open source exist for your software suppliers, systems integrators and outsourced developers as well. Organizations that carefully manage their own incorporation of open source may unwittingly incur unknown open source license obligations through code that has been introduced through outside software, firmware or hardware suppliers.

Open source software also requires special management because it is “delivered” with very different characteristics from commercial software. Open source typically comes “as is” with a disclaimer, not a warranty. An open source project may or may not have a structured approach to quality assurance (QA), in fact it may not have any QA at all! Non-commercial open source projects do not typically provide any support beyond the ability to post a question to a forum or mailing list. It is usually the downloader’s responsibility to keep track of patches and new releases, download and then integrate them. For many popular open source projects commercial service level agreement (SLA) support is available, but from independent providers, not the open source project itself.

Another important difference is that open source software is typically delivered without any kind of intellectual property indemnification, as is customary with commercial software. This means that the user bears all responsibility for any intellectual property claims that might come about through their use of the open source software. Again, it is possible to purchase indemnification insurance for many popular open source projects, but this must be purchased from independent providers.

Open source is licensed software, even though it is delivered through a frictionless, implicit acceptance process; this matters when it seeps into a company’s software products or infrastructure. Open source is made available under a wide range of licenses (over 400 at our last count), and the terms of some of these licenses may be incompatible with your company’s intended use. ***It is no more sensible to incorporate open source without review of its license that it is to sign a commercial software license agreement without reading it.*** Even where open source licenses are completely compatible with a user’s intent, the license may contain obligations that must be tracked and managed over time.

As a result of all of the factors sighted above, the companies that use and manage open source well have instituted special policies and processes for acquisition and management of this software. We have seen companies that do not manage their use of open source encounter serious issues such as:

Discovery of license violations in redistributed code

License violations that are discovered ‘too late’ often result in unforeseen (and unnecessary) legal expense. Costly injunctions to stop distribution of a product can occur and unanticipated requirements to publish code that a company considered proprietary can be the result. In many cases required remediation results in unexpected development costs and product shipment delays. Some companies have even incurred public embarrassment that has affected their business.

Operational problems due to untracked open source software

We have seen companies incur unexpected development expense and significant delays after they incorporated low quality open source code. Problems are especially likely where the companies did not have a clear and closely followed open source code review/update process in place. Some companies have had to take over the full burden of support, documentation and maintenance when they acquired code from moribund projects. We have often found serious bugs and security holes remaining in a company’s code base where no one is tracking posts and updates from the open source project. This is not to say that all or even a large portion of open source is bad and its use is problematic; rather we believe that planning and execution are vital to healthy software development, regardless of whether it is open source or proprietary.

Problems introduced through outsourced and contracted development

Companies can unknowingly incur licensing and operational risks when contract developers or systems integrators incorporate unreported open source. Many times, these companies themselves are unaware of the open source they have incorporated into their products. Often a thorough analysis of the company’s code base is needed in order to discover the thread of licensing obligations. A proactive plan is the only way to avoid these licensing issues.

Problems with corporate acquisitions

Open source software can affect corporate acquisitions for both the “acquirer” and the “acquired.” We have seen companies suffer serious “haircuts” in their acquisition price (as much as 40%) when unreported open source software was discovered in their code base during due diligence. At this point companies being acquired should expect rigorous code scanning as part of the diligence process of major acquirers. We have also seen companies discover problems with code bases they had already acquired. In some cases they felt that they had paid proprietary software prices for free open source software, and in other cases they may have suffered from the kinds of licensing or operational problems identified above.

Example

In order to understand the potential complexity of the issues, let's look at a real world example. Broadcom, a producer of semiconductor solutions for broadband communications, contracted with an offshore 3rd party to develop some software drivers for its communications products.

These drivers were licensed to Linksys to be included in their wireless products. Cisco later acquired Linksys for approximately \$500 million.

Upon conducting post-acquisition due diligence, Cisco discovered open source code licensed under the GPL in some drivers written by the offshore 3rd party for Broadcom and acquired by Linksys. At this point the [Free Software Foundation](#) became involved and attempted through private negotiations an enforcement order requiring Cisco to make public the open source code in question.

After long months of negotiations Cisco agreed to open source the specific libraries and make them available on their website. Luckily for Cisco this code has little real intellectual property value, but the lesson learned is that there is significant potential impact to revenue streams when proper open source due diligence and code management practices are not established and adhered to.

FSF Resolves Cisco Linksys Infringement

"In late 2002 and early 2003 the Free Software Foundation (FSF) started seeing a pattern of license violation reports concerning a whole distribution that was part of a system. From the FSF perspective, the "whole distribution" represented a kernel (Linux), a C library (glibc), the bash shell, iptables/netfilter, and a web server for configuration and the like. Because the typical whole distribution was predominantly GPL software, multiple potential copyright infringements and claimants existed.

"In such situations the FSF intervenes not only to defend its own right, and the right of public access to source code, but also to aggregate claimants. In this way, the allegedly infringing party receives assurance that by settling with the FSF, it will not face claims from anyone else. Because multiple parties apparently received their free software from the same upstream supplier, the FSF hypothesized that the trail would lead back to one originator, the chipset supplier for the various OEM systems.

"Aggregating the claims served another important purpose. It brought everyone's attention to a common problem: regulators around the world, nervous about software-controlled radio, were effectively bullying suppliers of such appliances into violating the GPL as a means to maintain the drivers as proprietary intellectual property. The FSF made the point to regulators, that aside from the GPL license infringement, allowing the community to develop and improve the drivers would be better business for the suppliers.

"Seeking to bring everyone to the table by the shortest route, FSF picked Linksys, now owned by Cisco, as the target of an initial enforcement effort. Cisco readily agreed to put a written offer for source code to the free software in the packaging, and to take some other basic and comparatively inexpensive remediation measures. Broadcom, the upstream supplier of almost all the software, opened direct discussions with FSF. In addition to the licensing resolution they wanted assurance that the regulatory quandary was understood and addressed. FSF agreed to support Broadcom in making sure that all other issues of source release were solved, enabling Broadcom's downstream OEM customers to deal on compliant terms with their customers as well."

--- Eben Moglen, General Counsel,
Free Software Foundation

The benefits of managing open source software well

Companies that properly manage their use of open source software experience clear benefits. The most obvious advantages lie at the heart of the open source phenomenon itself: it is much, much quicker and less expensive to use free, working code than to write and debug it yourself from scratch. This is especially true in new areas of technology where the pioneering work may be developing in an open source community or where a project involves interoperability with Linux or other major open source software systems.

Companies that take a best practices approach also experience other operational benefits. Through their management processes they insure that they match code characteristics to their intended use, e.g. they select mature, well supported software for mission critical systems, but encourage use of bleeding-edge technology for prototypes. The best open source software governance programs insure that they track updates and maintain the open source they incorporate, procuring external support where it is appropriate or actively engaging with the community for support. Best practices also encourage and support open source code re-use within the organization, making the most of expertise and consolidate in-house tracking, support and maintenance activities. In areas where leading edge development is occurring in open source communities, companies with a visible, well-run open source program experience an edge in hiring the best technologists. We are seeing many of our clients designate staff to work closely with open source communities and even contribute significant amounts of code. This engenders community goodwill and yields tangible benefits such as input into the direction of the project and functionality of the code, early awareness of changes and improvements and the support of the community for long term maintenance.

Finally, companies that manage their open source well are also managing their legal risks by reviewing license rights and obligations against their intended use of each piece of open source software. Where it is appropriate and cost effective they can acquire IP indemnification from 3rd parties. By tracking open source use and deployment they insure their compliance with license obligations.

The companies that manage their open source well are also more prepared for acquisitions. When being acquired they inspire confidence and secure their valuation by presenting a clear and complete picture of their open source IP use, licenses and compliance. When acquiring other companies or software, the policies, processes and tools are in place to determine what is in the software about to be acquired and what must be done to bring it in line with their own requirements.

Fundamentals of open source software management

Companies that manage open source software well tailor their management approaches to fit with other policies and processes already in place to serve the company's business strategies and objectives. However, the fundamentals of open source management are always the same.

Companies with the best practices:

- Know what open source software they are using and its essential attributes
- Know where they are using each open source component and that its use is appropriate
- Know who is responsible for the maintenance of each component
- Know that they are complying with their open source license obligations.

Know what open source software is in use and its essential attributes

Best practice companies keep careful track of the open source code they are using. In addition to the origin of the code, they also maintain their assessment of the maturity of the code, and the level of support available from the community or other sources. It is also essential to track the license under which the code was acquired. This is especially important for vendors whose rights to redistribute products incorporating the open source code are defined by the original license.

Understand where each open source component is used

In general, the suitability of a given component of open source software is dependent on where and how it is used. For instance, a newly created component with no community momentum behind it might be fine for an innovative prototype, but probably not for a mission critical system. A component under a “viral” license is acceptable for internal use, but not when combined with proprietary code in a product distribution. In addition, it is operationally important to know where a given open source component is in use when bugs must be fixed or security risks must be evaluated.

Identify who is responsible for the maintenance of each component

Unlike traditional commercial software models, open source software places most of the burden of support upon the user. For this reason it is critically important that each open source component in use have a designated “owner” responsible for keeping abreast of bug fixes, security bulletins and upgrades related to their component. The consolidation of these activities for multiple users of a component within a company provides important incentive for reuse and can result in significant economies. In cases where commercial support is available, the support provider may act as the maintenance owner.

Complying with open source licenses

Insuring compliance requires checks and audits. Companies with best practices employ processes that check for license compatibility before a component is incorporated for a particular use and then audit for obligations before production release or product deployment. When compatibility standards are clearly laid out in an open source policy the up-front decisions are typically efficient.

Best practices for managing open source software

Companies that employ the best practices in open source software management maximize the development productivity advantages of open source, minimize the operational and legal risks and accomplish this with very low overhead once the management programs are established. Although each company must tailor its open source management practices to work efficiently with its other policies and processes, Olliance Group has found that the following elements are always part of the best practices.

Assessment

Companies looking to establish an effective and efficient open source management program must start with a clear picture of their current and planned use of open source software. Sometimes it is also important to understand how their most important competitors are using open source, as well.

We have found in our work that most companies that do not have formal open source management programs greatly underestimate the degree to which they are using open source software and its potential impact. This has been equally true of companies that believe they have a grasp of the use of open source within their organization. Many companies that have told us “We do not use open source” or “We only use open source in this one area” have found through assessment that they depend on scores, sometimes even hundreds, of open source components in their software base. A management practice that is not based on reality is far worse than ineffective - it can undermine confidence and discipline within the organization and expose the company to numerous financial and legal risks. For this reason we recommend that companies new to open source management start with an assessment to understand in detail where they are using open source software and why.

Strategy

In order to develop and maintain an effective approach to open source management a company must understand at the executive level why it uses open source software to achieve specific goals and where it does not use open source and for what reasons. A strategy document is typically the tool for developing consensus among executive stake holders such as software development, production IT, line of business management, procurement and legal. Once it has been developed, the strategy document is a key educational resource for the open source management program – people implement programs far better when they understand the goals and rationale behind it.

Policy

An open source policy is at the heart of any open source management practice. It clearly and succinctly specifies the criteria and decision points for use of open source, the information that must be collected and tracked, and the roles and responsibilities for managing it. The policy document itself must be simple and brief enough that every developer will read it and every manager involved with open source will know it, remember it, and follow it. The decision criteria in the policy should be sufficient to decide at least ninety percent of the open source use cases, otherwise exceptions will bog down the management processes.

Process

An open source policy is merely a dead document unless it is coupled with a management process, and the effectiveness and efficiency of the open source management process are critical success factors in any program. Developers need timely response to their requests to incorporate open source or productivity is lost. Exceptions to the policy must be considered, but in a way that makes efficient use of scarce management time. Good open source management processes must be reasonably well integrated into a company’s existing software development tools and processes. The most efficient processes that we have seen make use of on-line workflow applications – the benefits are particularly pronounced in large or geographically distributed organizations. Several open source management workflow tools are now commercially available.

Repository

Good decision making depends on the ready availability of good information. Effective open source management requires keeping track of open source components, their attributes (including their licenses and where the code was obtained from), their owners in the organization, where they are used, and incorporating decision, modification and maintenance histories.

Success factors for open source IP management

In addition to the essential elements described above we have found several factors that are always present in the most successful open source management programs.

Development ownership

Because open source software is primarily about software development, management programs that are visibly championed by software development management leaders are much more likely to be effective and efficient. In particular, software developers are much more likely to comply with the company's policy and process when their managers visibly support the program. We have observed that active participation by engineers in the company's open source management program is key to its fundamental objective of boosting productivity.

Training

Companies that train all participants in its open source management programs achieve much higher compliance and better efficiency. Adequately training for development engineers is particularly important, because they are the front line of implementation. If these engineers don't know and understand the policy, they cannot follow it. If their understanding of the policy is vague, mistakes and unnecessary exception handling will result. Our experience is that engineers are even better decision makers when they understand the objectives and rationale behind the policy (see Strategy above).

Tools for information sharing

As with any human process communication is an important success factor. Companies with the most effective and efficient open source management programs employ several types of information sharing tools. The most important is an on-line workflow application to support the process. In addition to managing the routing of requests and approvals the best programs encourage open source reuse by providing useful information about already approved software and previous decision making. Best practices companies also utilize portals, forums and FAQs on their intranet. These tools help programmers find appropriate open source (or the company's own code) and improve program efficiency, communication and a sense of ownership and community in the process.

Summary and Conclusion

Open source software is copyrighted and licensed intellectual property that requires different management techniques than traditional proprietary software due to its free-wheeling distribution, its different license structures and its self-service support and maintenance conventions. Companies that manage their use of open source software realize substantial productivity benefits; those who do not incur operational, financial and legal risk. Fortunately over 25 years of open source distribution and use have yielded a set of proven "best practices" for its management.

Companies employing best practices in open source management track what open source they are using, where they are using it, how they are maintaining it and their compliance with license obligations. They do this by employing a clearly articulated *strategy*, a clear and concise *policy*, an efficient *process* and a tracking *repository*. The best-in-class programs employ technical, operational and business leadership, regular training and on-line productivity tools.

Implementing an effective and efficient open source software management program is an art. Adept skill is required to integrate a company's strategy and priorities across software development, production IT, legal, procurement and line of business management and to communicate the details and rationale of the program elements effectively. Finding individuals with the requisite knowledge and experience spanning commercial software development, open source communities, licensing and legal risk management and operations can be a serious challenge for most organizations. Fortunately, qualified resources and time-saving tools are now available to help companies ensure their success in realizing the productivity benefits of open source software while effectively managing the risks.

About Olliance Group

Olliance Group is the leading independent open source business and strategy consulting firm. Our mission is to help clients capitalize on the strategic, technological, and financial benefits of open source software. Clients include Fortune 100 enterprises, independent software vendors (ISVs), start-up firms, venture capital groups, not-for-profit organizations and government entities.

With the collective team experience of more than 150 years of open source business strategy, and our independent position in the open source ecosystem, we are able to offer a unique and powerful set of services including:

- corporate open source strategy
- product and technology strategy
- community development
- sales and marketing
- planning and implementation
- IP management and compliance
- financial and mergers & acquisitions planning.

During the last seven years, Olliance has completed over 250 engagements with high profile entities including IBM, Nokia, Intel, Microsoft, Sun, Wells Fargo, Bank of America, and the US Navy. Olliance Group has built a strong reputation in the venture capital community as well as with more than 30 commercial open source startup clients, seven of which were named as top 10 open source business models to follow by industry analysts The 451 Group.

Olliance plays an influential role in the open source industry and to community organizations through our strategic engagements and in maintaining a direct connection to the open source community at large by participating in a number of open source projects and user communities.

Olliance Group's open source IP Management and Compliance Practice helps clients understand the value and impact of open source software and develop strategies and policies that realize the productivity benefits while managing potential risks.

For further information please contact:

Norm Swaton
Marketing Manager
nswaton@olliancegroup.com
650-493-3800 x 214
www.olliancegroup.com