

Practicing Safe Software

by Sorin Cohn-Sfetcu, Richard Mayer

Intellectual Property Compliance and Protection via Tools that Automate Software Record Keeping and Create a Software Bill of Materials

You are probably aware of the recent decision of the US Court of Appeals for the Federal Circuit and might now be wondering about what's in the software that's so pervasive and critical in your IT network. Yes, using third-party software – either open source or outsource developed – may lead to Intellectual Property (IP) copyright infringement issues, but rest assured the problem is manageable, especially with the latest tools now available in the market.

The US Court of Appeals ruling sharpened the claws of open source software licenses, like CC and GPL, which set conditions on their use. If you violate the conditions, the license disappears and you become a copyright infringer with an *opportunity* of landing in court. Does this mean that open source software is detrimental? Not at all – as long as you know what software components are in your code and you abide by the licensing that rules them.

Open Source Software – The Good (& Profitable)

The software industry has matured and software has become an ubiquitous part in many products – especially so in telecommunications and IT. Why? Simply because it enables faster development, short introduction intervals and lower development costs.

There are thousands of modules and millions of lines of code available, and outsourced software development is prevalent in the industry. Call processing, network and performance management, almost everything is available from open source projects like *Asterisk*, *OpenSIPStack*, *FreeSwitch*, *RTPLib*, *SipX*, *OpenPBX* and so many others.

Software is pervasive, and software code is now treated as a commodity component very much like electronic components or nuts and bolts are treated in hardware industries. One major difference sticks out: software code is easily taken, transported and incorporated, but becomes laden with licensing and IP issues.

Open Source Software – The Bad (& Problematic)

Growth of outsourcing and open source with hundreds of license variations and complexities of software practices, make code contamination a very real and common occurrence. These days, software development has become an assembly of files and code modules some of which may be new and proprietary, while some may come from open source repositories, from outsourced developers or from commercial sources. Re-using code from these various sources is important for profitability and efficiency, as costs are paramount and time-to-market is critical. Re-inventing a common function in software can be tedious and frustrating to a creative developer.

As shown in numerous legal and business instances in today's world of open-sourced, out-sourced, easily-searched and easily-copied software – quality development coupled with careful managing and safeguarding of IP is critical. IP contamination or even perceived IP issues can delay projects, product sales cycles, or business transactions, while reducing the value of the software itself or even of the entire company. Worse, it may lead to onerous IP infringement rulings.

Note the case of Veritas vs. Microsoft, where the *Court noted that even where a relatively small quantity of code is copied,*

a finding of substantial similarity can still be made if the copied code is sufficiently important to the operation of the new program. In this famous case, the *Court observed that only 54 lines of code, or 0.03% of a code base of almost 160,000 lines, had been identified by Veritas as having been infringed.* In addition, with the exception of two lines, the section of code in question was not copied verbatim. Instead, Microsoft changed the code by upgrading the programming language from C to C++. (Veritas versus Microsoft, United States district court, Western district of Washington at Seattle, Case NO. C06-0703-JCC).

Open Source Software is Not Free

Commercial as well as open source software is governed by license regulations and restrictions. Commercial licenses can be evaluation licenses and can be affected by how often or how the code is used, can have export restrictions for political or security issues, or can be limited for use only in certain types of applications.

While open source software is freely available, it is not free of regulation, as it is associated with licenses that affect use, modification and distribution. The more permissive licenses typically allow use as needed and have limited rules or conditions attached. An example is the BSD (*Berkley Software Distribution*) license. There are also restrictive licenses like the GPL (*General Public License*). The most common restriction is known as *copy left* which imposes contribution back to the open source and full publication of any derivative works. There are many different open source licenses that span the spectrum of restrictions, some of which may lead to license incompatibilities. Even hosted

services may become affected by open source license restrictions.

What's worse, license terms may be nested through several levels of software code and sometimes may be incompatible. The license terms are difficult to interpret and require either legal intervention or expensive training of developers and product managers who would rather spend their time and intellect on more engineering tasks. The result?

Few People Know What's in the Software

The result is that nobody knows for sure what's in the final software load. There is little pedigree information: how the software evolved and from what; who brought in what; who owns the IP for the code, or what license pertains to that piece of software. Record keeping, if diligently followed at all, is at best manual; i.e. impractical, subject to error, time consuming and an unappreciated task.

Software development practices have evolved over time to include systems for checking syntax, managing software versions and tracking software bugs, but some key disciplines that are standard in hardware development have yet to be adopted for software:

- An approved vendor list containing the qualified components and licenses for developers to select components freely from the list without concern,
- A Bill of Materials (BoM) to enable proper use and distribution, determine value and track vendor upgrades and other post-design activities.

Managing Open Source Software: Leverage the Good and Eliminate the Bad

Discipline in software governance requires that:

- Clear software IP policies are defined in line with organizational goals,

- The supplier list is rationalized for cost, performance and maintenance efficiencies,
- Accurate records are kept.

A sound software IP license policy is clear and provides an effective list of approved vendors and acceptable types of software to ensure IP compliance and safeguarding:

- What is allowed and what is restricted,
- How often a particular component could be used and for what purpose,
- Goals and guidelines for re-use,
- Potential use and distribution restrictions, including export restrictions – which software goes into which product sold into which country.

Setting up a software license policy is not trivial, as it requires educated input from legal, business and technical perspectives. Ideally, such policies are centrally defined, should be easy to enforce and should allow tracking compliance as development proceeds apace. Naturally, a large organization with a variety of software projects must ensure that each project has its own appropriate software license policy.

Maintaining accurate records requires:

- Keeping track of all external and internal components in a project – open source, commercial or outsourced,
- Keeping a record of who brought in what and when and associating each piece of code its licensing and copyright attributes.

The result? A software BoM which becomes a critical indicative of the *Software Capital* of the enterprise.

Final Advice: Invest in Automated Record Keeping & Software IP Management Tools

If you are buying software, make sure it has a clean BoM. If you are selling it, ascertain beforehand that it has a proper pedigree to

avoid going to court or having to suffer through a retrospective due diligence process.

Several commercial software tools are available to supplant the archaic manual methods of ensuring software IP management, notably *Protecode*, *Palamida* and *Black Duck*. The last two companies focus on analyzing already developed software code, giving less attention to forward issues of development costs and time-to-market. Protecode has taken the approach of being both proactive, automatically tracking code as it is being brought in, as well as retrospective in analyzing legacy code and providing an overall software BoM. Protecode can be purchased either as a licensed software tool or a software service, with full confidentiality and security provided.

In general, such tools require little, if any, additional training and are scalable from single developers to multi-site, multi-organization teams allowing central establishment of IP policies and follow-up action on policy non-compliance. Best of all, they enable cost-effective, disciplined software development through automated detection, logging, identification, pedigree-tagging of software content, and provisioning of IP compliance reports and the associated software BoM.

Advanced automated software governance and IP management tools are becoming key business management tools for CEOs, CTOs, and legal officers who need them to reduce business risks, ensure IP compliance and protection and achieve higher revenues at lower costs with software intensive products and services.



About author

Co-authored by Sorin Cohn-Sfetcu and Richard Mayer, Protecode, Inc.
Email: info@protecode.com
Phone: 1-613-721-5936
www.protecode.com