


The 7 Myths of IP Risk: The Real Exposure Issues with Free and Open Source Software

Black Duck Software White Paper



FOSS is widely recognized as providing significant technology, innovation and financial benefits to software development organizations, but it is not without associated management challenges and risks. In this second paper of a two part series (Part One: **Managing the Operational and Security Exposure of Free and Open Source Software**) that examines key risk factors in the context of free and open source software (FOSS) use, we look at the intellectual property (IP) and legal risks that may be encountered when FOSS software makes its way into products, either in planned or unplanned use.

FOSS is pervasive in today's software lifecycle and supply chain. Gartner estimates that by 2013 open source technology will be included in 85 percent of all commercial software packages, and by 2016 will be included in mission-critical software packages within 99 percent of global enterprises¹. Clearly FOSS is being rapidly adopted. In terms of risk, Gartner predicts that 50% of Global 2000 organizations will experience technology, cost and security challenges through lack of open source governance by 2013. It's clear that FOSS is about opportunity, and yes, about risk, complexity, and management challenges as well.

In this paper, **IP Risks and Myths: The Real Exposure Issues with Free and Open Source Software**, we address the IP risks that developers, development managers, compliance officers, legal, and C-level managers must take into account when developing and distributing software created with FOSS in today's multi-source development processes. We also provide readers with a discussion of myths that surround IP exposure in FOSS and survey the real issues surrounding licensing obligations, IP exposure and good FOSS citizenship. We welcome your thoughts and comments.

Introduction

FOSS is licensed software, and with licenses come obligations

Software flows into development organizations today via many acquisition channels and through very different modes of payment. Three main types of software dominate enterprise IT: internally developed code; commercial software, which can include 'shareware', available for a fee and typically released under a restrictive license (Microsoft Windows, SAP

Business Suite, Oracle Database 11g); and FOSS, just a "mouse click away" and typically available free from public-domain sites (e.g., kernel.org for the Linux kernel, Mozilla.org for the Firefox browser, Fedora Project.org for the Fedora Linux distribution. It is also distributed under varying paid-support models by a few major vendors, e.g. Red Hat (Red Hat Enterprise Linux, RHEL), Canonical (Ubuntu desktop Linux), and Oracle (Oracle Enterprise Linux, a copy of Red Hat's RHEL) as open source.

Let's break down the categories a bit further and show some of the critical differentiators enterprise IT organizations using FOSS should know:

FOSS

Free and open source software, while free, does always come with a license; someone (often many individuals) retains copyright ownership over the source code and often trademark control over the broader project.

- FOSS licenses have obligations that must be met
- FOSS licenses are perpetual
- Some FOSS vendors depend on dual-license models that encourage users to 'try before they buy', as with shareware (see below), but users are never compelled to upgrade to enterprise commercial editions (one reason: FOSS code can always be forked.)

Shareware

Shareware encompasses a wide range of software solutions that focus on "try before you buy" or volunteer commercial contributions (often both elements). However, unlike public-domain code and more similar to FOSS, shareware is generally distributed under a license. In many cases, potential adopters can use the software freely (gratis), purchase and even redistribute it, but cannot modify it for their own purposes.

- Shareware and FOSS are licensed, but FOSS specifically provides the freedom to modify the code as well.
- The terms and conditions for use of shareware are often restricted as well — for example, for a 90-day trial period.
- Shareware often locks certain features unless you pay to access them.

FOSS - unique for its freedoms

Unlike Shareware or commercially-available software, FOSS offers the following freedoms to its users²:

- The freedom to operate (that is, run) the software, for any purpose.
- The freedom to study how the underlying program works, and change it to make it do what you wish — that is, examine and modify the source code.
- The freedom to redistribute copies to others.
- The freedom to distribute copies of your own modified versions to others.

² <http://www.gnu.org/philosophy/free-sw.html>

³ Various, Gartner Predicts 2011: Open-Source Software, the Power Behind the Throne. ©2010 Gartner, Inc. and/or its affiliates.

Most open source projects are built around communities of "contributors" - individuals who contribute code, patches, and updates to FOSS projects. Community-centric projects represent the traditional FOSS model, and many are entrenched in the ethic (open, free-of-commercial-oversight innovation) of FOSS. Many projects enjoy wide community support (or corporate sponsors, e.g. Android/Google), and many others have smaller communities and few users. Factors such as copyright control or IP governance vary dramatically among projects as well.

FOSS is abundant - hundreds of thousands of projects are available through thousands of repositories - and it has become mainstream, to the point that Gartner's lead FOSS analyst, Mark Driver, recently observed, "As OSS has matured, it has expanded its role within mainstream IT organizations to take on increasingly complex and mission-critical challenges. In 2010, Gartner estimated that open source is included in at least 75% of Global 2000 enterprises."³

"Open source is ubiquitous, it's unavoidable....having a policy against open source is impractical and places you at a competitive disadvantage"

-Mark Driver's take on the futility of fighting vs embracing FOSS use

Nevertheless, many enterprise IT organizations attempt to ban its use or are loath to use (or admit they use) FOSS, taking a 'head in the sand' approach. In part this is due to the 'myths' that surround FOSS.

The Myths of FOSS

A number of ‘myths’ surround FOSS. This group of myths comes from Karen Copenhaver, legal counsel for the Linux Foundation, and partner at Choate, Hall and Stewart. Some of these myths have held back its enterprise use. They include:

- You cannot use free and open source software in a proprietary environment (or you will die)
- All open source licenses require the release of source code for everything
- The easiest answer to free and open source is to just ‘say no’
- None of these (license and IP) agreements are enforceable so it doesn’t really matter anyway
- No one will ever know (if we use it but don’t observe license compliance)
- Our corporate policy says we don’t use FOSS
- We aren’t distributing software, so we don’t have to pay attention, right?

Let’s explore the myths and debunk them.

Myth #1: You cannot use free and open source software in a proprietary environment (or you will die)

Plenty of FOSS users are around and thriving. In fact Gartner states, “By 2014, those organizations with effective, open-source community participation will consistently deliver high returns from their open-source investments.”⁴ Whether this use is direct and intentional, or hidden and unintentional, is not important to debunk this myth; the point is that FOSS is in use in enterprises everywhere, behind the firewall and in products being distributed to customer and partners.

Myth #2: All open source licenses require the release of source code for everything

There are more than 2,000 licenses in use for FOSS, although fewer than 70 are approved by the Open Source Initiative’s (OSI) license review process. Each license differs in its terms. Licenses vary by type, reflecting the intent of their author(s). The choice of a license by a project usually reflects the project

leader’s intent for the use of its software. Not all licenses require the release of source code; restrictive licenses may, but many permissive licenses do not.

License types include:

Restrictive (aka Copyleft, reciprocal)

- Requires licensor to make improvements or enhancements available to all under similar terms
- An example is the GPL: Licensees must distribute “work based on the program” and cause such works to be licensed at no charge under the terms of the GPL.

Permissive

- Modifications/enhancements may remain proprietary
- Distribution in source code or object code is permitted provided copyright notice and liability disclaimers are included and contributors’ names are not used to endorse products
 - Examples: Berkeley Software Distribution (BSD), Apache Software License.

Miscellaneous licenses

- Examples: Zlib/libpng.

Myth #3: The easiest answer to FOSS is to just ‘say no’

It’s easy to say ‘no,’ but if you’re the head of development of a global software organization with a few thousand developers, various third-party software suppliers and numerous ISVs, that ‘no’ won’t carry very far. The reason? FOSS comes into your enterprise from multiple sources, e.g.:

- Embedded in closed-source commercial applications (downstream liabilities exist)
- From internal developers with browsers (anyone with a browser has the ability to search for and download code with license obligations)
- From outsourced/offshore/nearshore software product development organizations or captives
- Via acquisitions and mergers.

Saying ‘no’ doesn’t prevent the influx of FOSS, and it can lead to denial, which may open your organization to significant risk. Again,

⁴ ibid.

from Mark Driver of Gartner: “Whether directly adopted within in-house IT efforts or acquired indirectly via an independent software vendor (ISV) or a packaged application channel, open source will continue to coexist with closed-source solutions in a blended (i.e., hybrid) presence among mainstream IT solutions in mission-critical roles. IT organizations must clearly understand where and how they are using OSS.” (Emphasis is ours.)

The final point here is that not only is saying ‘no’ impractical from a prevention perspective, depriving your developers of leveraging the significant technical and financial benefits of FOSS will put them at a competitive disadvantage. (A subject for another paper perhaps but there’s also the issue of if you plan to hire developers out of college, most of whom have made extensive use of FOSS during their education, having a policy against using FOSS will severely limit or prevent you from hiring good talent).

Myth #4: None of these (license and IP) agreements are enforceable so it doesn’t really matter anyway.

Where there’s a license there’s law, and a lawyer or organization (Groklaw, Free Software Foundation, Software Freedom Law Center) willing to prosecute violations, and often in support of the FOSS developer community. The recent Jacobsen v. Katzer case is a cautionary tale that busts Myth 4. Jonathan Moskin and Howard Wettan, two attorneys at White & Case LLP report that, although leaving some questions unanswered, last summer’s court decision in this case “held that breach of an open source license does not merely permit a breach of contract claim, but that violating the ‘conditions’ to the intellectual property license creates a cause of action for copyright infringement -- with associated [financial] remedies....An additional consequence of the Federal Circuit’s opinion was that, by pursuing copyright claims, an open source licensor may now be able to sue downstream licensees for copyright infringement.

This case, which generated a great deal of press regarding the downstream liabilities that can be hidden in FOSS licenses, is illustrative. Attorney Moskin, cited above, was interviewed by Betanews. In the interview about the JMRI case, Moskin offered the example of a bank hiring a software developer to write a piece of billing software for use by its customers.

“Let’s say the developer downloads a piece of open source software. The bank thinks its code is proprietary, but it actually incorporates open source. Other businesses [using the billing software] might then also be liable for copyright infringement,” Moskin illustrated.

Count on FOSS licenses to be enforceable, and know that there are plenty of people - the software’s creator, the community, or an organization such as the Free Software Foundation - willing to enforce the terms of the license.

Myth #5: No one will ever know (if we use it but don’t observe license compliance.)

The above example proves this myth wrong, plus there are tools designed to find violations (<http://www.binaryanalysis.org/en/home>) and organizations that routinely look for and report violations, including www.gpl-violations.org. A few years ago this myth may have been more common, but with the advances in technology for automatically discovering FOSS, the increased awareness of FOSS and its associated obligations, it’s much less so today.

Myth #6: Our corporate policy says we don’t use FOSS.

Corporate policies are made to be broken - largely because so few employees know what they are - and it’s not just the developer group that thinks so.

Developers working on a programming challenge look for the best code to solve the problem. Often time constraints and the



nature of the work lead to a search for code that will match the need: Who needs to write another parser? Why reinvent the wheel? Your competitors likely are using FOSS to advantage, can you afford not to? A quick Internet search will turn up many FOSS projects - Black Duck tracks more than 475,000 from more than 5,000 unique websites in its KnowledgeBase. It's important to have a FOSS use policy, but the only way to really be sure what's in your code is by using technology to automatically identify, manage and control the FOSS and other code in your code base.

Myth #7: We aren't distributing software, so we don't have to pay attention, right?

With over 2,000 software licenses being used, license obligations vary. With some, license obligations exist independent of whether software is distributed to customers, partners, and associates. Large Enterprise IT organizations, utilizing 100s to 1000s of FOSS projects, need to be aware of license obligations to manage compliance and risk.

An added wrinkle: while you may not be developing the code for redistribution, mergers and acquisitions can change everything. If your company is considering a change in status, it's critical to know what's in the code you're acquiring before final agreements are inked. After all, your IP - including all software - is a significant asset.

Conclusion

FOSS is pervasive in today's software lifecycle and supply chain, and its use is now a mainstream strategy for many development organizations. But while its use has become pervasive on the part of development organizations, management and control has lagged far behind. FOSS is free but it must be managed as an asset and license obligations observed. In addition, FOSS often carries with it the obligation of contributing back to the community. The Enterprise IT organizations that succeed with FOSS will be those that move beyond common cause to what Gartner has labeled 'collective competency'⁵.





About Black Duck Software

Black Duck Software is the leading provider of products and services for automating the management, governance and secure use of free and open source software, at enterprise scale, in a multi-source development process. Black Duck® enables companies to shorten time-to-solution and reduce development costs while mitigating the management, compliance and security challenges associated with free and open source software. Black Duck Software powers Koders.com, the industry's leading code search engine for open source, Ohloh.net, the largest community for and free public directory of open source, and The Olliance Group, the leading open source business and strategy consulting firm. Among the 400 largest software companies in the world, according to Softwaremag.com, the company is headquartered near Boston and has offices in San Mateo, California, London, Paris, Frankfurt, Hong Kong, Tokyo and Beijing. For more information, visit www.blackducksoftware.com.

© 2011 Black Duck®, Know Your Code®, Ohloh®, SpikeSource®, Spike®, and the Black Duck logo are registered trademarks of Black Duck Software, Inc. in the United States and/or other jurisdictions. Koders™ is a trademark of Black Duck Software, Inc. All other trademarks are the property of their respective holders.



Contact

To learn more, please contact:
sales@blackducksoftware.com or
call +1 781.810.5100

Additional information is available
at Black Duck's web site:
www.blackducksoftware.com