SYRACUSE SCIENCE & TECHNOLOGY
LAW REPORTER

VOLUME 22 SPRING 2010 ARTICLE 4, PAGE 79

Open Source or Open Season?: What legal professionals need to know about open source software before dealing in corporate transactions and the ramifications of GPLv3

Emily Prudente*

INTRODUCTION

Open source software, until recently, was a relatively unknown part of the landscape of software development. With the Internet's growing ubiquity and usage, the use of open source software increased and evolved rapidly. The unique licensing schemes covering open source software make it particularly intriguing from a legal perspective, and for uninformed legal professionals. Without an acute understanding of the details of these licenses the world of open source code quickly becomes treacherous and ripe with potential pitfalls. Open source licensing creates liabilities and with its overwhelming benefits and increasing acclaim, this new software is impossible to ignore.

Software developers and copyright holders have begun to turn to the courts for guidance on the degree of recognition and level of protection they can expect from the judicial system.[1] Recently, there have been a number of actions addressing the enforceability of open source licenses.[2] While most suits have settled, there has been important litigation that suggests how the legal system is likely to analyze cases involving these licenses.

---

* J.D. candidate, Syracuse University College of Law, expected 2010; Lead Articles Editor, *Syracuse Science & Technology Law Reporter*.

[1] Rachel Stern & Erik C. Kane, *Open For Business: What Corporate Counsel Need to Know in the Intricate World of Open Source Code*, ACC DOCKET 26 NO. 10, 46 (2008).

In the context of mergers and acquisitions, recognition and management of open source code is critical for a number of reasons. The presence of open source can affect the value of a company's intellectual property assets, thereby affecting the value of the deal itself.[3] The risk of lawsuits and lost revenue increases when upper management or in-house legal counsels are not kept abreast of the type of licenses that may be intertwined with their proprietary software.[4] For attorneys practicing corporate law, intellectual property law, or those who serve as general counsel to technology companies, awareness of the increasingly complex world of open source licensing is of paramount importance.[5] Companies stand to lose a great deal, in some cases, everything, if open source code is not properly managed. Despite the risks, however, open source software has established itself as an intangible asset companies can no longer afford to ignore and whose benefits far outweigh the risks.

Understanding the Basics of Open Source Software

A solid understanding of open source software is crucial to properly manage it.[6] To appreciate the value of open source software, it is helpful to begin by understanding the basics of computer software. The term "software" refers in general to computer programs that function to operate the computers and run various and sundry devices related to computers.[7] Common

---

[2] Jacobsen v. Katzer, 609 F.Supp.2d 925, 928 (N.D. Cal. 2009).

[3] Alan Stern, *Open Source Licensing*, Practising Law Institute: Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, 915 PLI/Pat 187, 196 (2007).

[4] *Id.*

[5] *Id.*

[6] Stern & Kane, *supra* note 1, at 42.

[7] Dr. Jose J. Gonzalez de Alaiza Cardona, *Open Source, Free Software and Contractual Issues*, 15 TEX. INTELL. PROP. L.J. 157, 162 (2007) [hereinafter Gonzalez].

practice among software engineers is to write programs in easily interpretable computer programming languages, such as C++ and Java.[8] When programs are written in these languages the resulting source code is comprehensible to programmers and engineers, but not to machines.[9] In order to execute a program the computer must translate the source code into what is known as object code, or binary code.[10] After translation, the source code is a series of 1's and 0's that only machines can interpret.[11] In the interest of maintaining and protecting trade secrets, engineers frequently deliver programs in object code form.[12] Reverse-engineering source code from object code is so difficult it is effectively impossible, with the result that if a program has been delivered in binary form, there is no reasonable probability that purchasers will uncover the underlying source code.[13]

Protection of proprietary code, particularly for companies that derive significant portions of revenue from their intellectual property assets, is vital to advance and thrive in their respective fields. If software is delivered by the developer in source code form, the company becomes vulnerable to disclosure of their trade secrets and exposure of its intangible intellectual property assets it has spent substantial time and money developing.[14] In instances such as these, a firm's intellectual property may be the cornerstone of its competitive advantage in the marketplace.

---

[8] Stern & Kane, *supra* note 1, at 42.

[9] Gonzalez, *supra* note 7, at 163.

[10] *Id.*

[11] *Id.*

[12] *Id.* at 164.

[13] *Id.*

[14] Stern & Kane, *supra* note 1, at 43.

But while protecting trade secrets can be critical to a company's survival, it comes at a price. The obvious goal of developing software for sale or license is to generate revenue by providing solutions that competitors are not able to offer to the market. In accomplishing this goal software is developed by engineers in all companies in an environment that promotes secrecy instead of in an atmosphere that encourages the sharing of ideas between developers. The philosophy of open source software is to overcome corporate barriers of secrecy in order to foster innovation more rapidly and efficiently among developers at different companies who share common programming goals.[15]

The concept of synergy is a pillar in the philosophy of open source code; the more mind-power devoted toward a common goal, the richer and more robust the end-product will be. In the eyes of those who are involved in the movement, the resulting whole can in fact be greater than the sum of its parts.

A number of forces have facilitated the movement for sharing open source code. Due to its increasing use and development, the internet serves as an excellent medium through which code can be more readily accessed and shared among peers.[16] Supporters of the open source software movement encourage sharing ideas and improvements when open source is made available to a community of peers. This support forms the foundation of the open source philosophy.[17] Also, there is the cost-effective benefit to using open source code because it is less expensive than proprietary, closed source code.[18] Free access to programmers' work product

---

[15] *Open Source Definition: Annotated*, Open Source Initiative, http://www.opensource.org/docs/definition.php.

[16] Stern & Kane, *supra* note 1, at 42.

[17] *See* source cited *supra* note 15.

provides programmers with the opportunity to incorporate their ideas into new program while allowing others to improve the code without the financial burden of licensing fees or royalties.[19] Of course, this depends in part on the license under which the source code is covered. These licenses will be discussed and analyzed in detail later in this paper.

Open Source Initiative (OSI)

The open source movement began decades ago and evolved from what was originally called the free software movement, which Richard Stallman began.[20] A brief discussion of free software is necessary to appreciate the subtle differences between free and open source software.[21] A common misconception is that free software is "free of charge." This is not true. The free software movement emphasizes the freedom to replicate, adapt and subsequently redistribute the work.[22] On the other hand, the open source movement emphasizes access to the source code but not necessarily the ability to modify and/or redistribute the program.[23] At first impression the open source philosophy may appear to run counter to the goal of most corporations to generate a profit. Bill Gates, a well-known opponent of open source posited, "One thing you do [when implementing open source philosophies] is prevent good software from being written. Who can afford to do professional work for nothing?"[24] Programmers who

---

[18] Gonzalez, *supra* note 7, at 171.

[19] Stern & Kane, *supra* note 1, at 42.

[20] *Id.*

[21] Gonzalez, *supra* note 7, at 167.

[22] Matt Lee, *What is Free Software and Why is it so Important for Society?*, FREE SOFTWARE FOUNDATION, March 29, 2010, http://www.fsf.org/about/what-is-free-software.

[23] Gonzalez, *supra* note 7, at 180.

participate in the open source movement are typically motivated by goals other than direct economic gain.[25] There is satisfaction and prestige in creating or modifying a program, and also the possibility of economic benefit from sources other than copyright.[26] The open source software movement is premised in part on the idea that with plentiful beta-testing[27] and with enough sets of eyes, any program can be improved and de-bugged.[28] Such a community-based approach, therefore, increases efficiency in the software development market.[29]

The open source, as opposed to free software, movement officially began in 1998 when Netscape announced it would release the source code for its popular internet browser.[30] The Open Source Initiative (OSI) was formed by a group of scientists whose goal was to make the open source concept attractive to the corporate world.[31] While some free software enthusiasts, such Linux creator, Linus Torvalds, support the open source movement, others like Richard Stallman, do not.[32]

---

[24] Gonzalez, *supra* note 7, at 177.

[25] *Id.* at 175.

[26] *Id.*

[27] The concept of beta-testing refers to the common practice of software engineers to trial, typically in-house, a version of their code before releasing the program for use either internally or for external customers. Essentially, it is a quality control mechanism that offers some assurance of reliability for a brand new product.

[28] Gonzalez, *supra* note 7, at 176.

[29] *Id.*

[30] *Id.* at 178.

[31] *Id.*

[32] *Id.* at 179.

Prevalence of Open Source

There are successful for-profit corporations whose business models are built around open source, such as Red Hat.[33] Red Hat is famous because it is built on the model of "media distribution, branding, training, consulting, custom development, and post-sales support" and delivers both software and maintenance support for its patrons.[34] Many products based in full or in part on open source code are increasing in prevalence. For example, almost 70% of web servers are run by Apache, an open source software package.[35] Many domain name servers, like Mozilla's Firefox, implement BIND which is another open source package.[36] Outside the cyber-world, antilock brakes, watches, mobile phones, PDAs, television set-top boxes, and medical equipment are just a few of the electronic devices that incorporate open source software.[37] The open source Linux operating system, based on open source code, has roughly 29 million users.[38] This market share is dwarfed by the number of users of the widely-known Microsoft Windows operating system.[39]

The GPL

Of all the licenses covering open source products and services, the most popular is the General Public License, or, commonly, the GPL. Linux, Playstations 2 and 3 and some cell

---

[33] Gonzalez, *supra* note 7, at 177.

[34] *Id.*

[35] *Id.* at 160.

[36] *Id.*

[37] *Id.* at 161.

[38] Gonzalez, *supra* note 7, at 161.

[39] *Id*.

phone carriers all are licensed under the GPL version 2, which, until recently, was the newest version.[40] The idea for a general public license emerged from the difficulties inherent in placing computer programs in the public domain. Allowing access to open source software without any legal structure would lead to complications that would effectively frustrate the community-oriented goals of free software.[41] For example, programs in the public domain may be released in object code form.[42] Also, without licensing restrictions, even those programs released in source code could be redistributed only as object code.[43] Releasing programs in object code format prevents downstream users from accessing, adapting or redistributing the program. This format interferes with the sharing of the ideas, and that concept is central to the open source philosophy.[44]

The GPL, also called the GNU GPL, is issued to the majority of users of open source software.[45] The license has two primary clauses. The first grants users the right to access, modify and redistribute the source code. The second requires that redistribution be performed under the same terms as those of the license that gave permission to access, modify and redistribute the code.[46] This second part is known as the "free software clause" or "copyleft clause" because it permits redistribution without authorization and thus denies the creator one of

---

[40] Stern & Kane, *supra* note 1, at 42.

[41] *See* source cited *supra* note 22.

[42] Gonzalez, *supra* note 7, at 173.

[43] *Id.*

[44] *See* source cited *supra* note 15.

[45] *Id.*

[46] GNU General Public License, version 2 (2001), http://www.gnu.org/licenses/old-licenses/gpl-2.0.html#SEC3.

the exclusive rights held by copyright owners, the right to distribute.[47] The GPL is rendered truly complex and unusual by the inclusion of the copyleft clause.[48] The perpetual nature of the freedom to alter software by downstream developers gives the GPL a viral, reciprocal, or hereditary, characteristic.[49]

The significance of the copyleft clause is that any open source program, whether modified or not, must be redistributed to downstream users with permission to access and modify the code.[50] Copyleft licenses impose an obligation on downstream licensees to redistribute their respective programs – modified or not – under the same copyleft clause.[51] This treatment of modified software that has been redistributed is the critical difference between open source licenses and proprietary commercial licenses.[52] It is illustrated by a comparison of two of the most notable open source licenses: the GPL and Berkeley Software Distribution (BSD).[53]

---

[47] The GPL states, "You may convey a work based on the Program, or the modification to produce it from the Program, in the form of source code . . . provided that . . . c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply . . . to the whole of the work, and all its parts, regardless of how they are packaged." GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html.

[48] Sapna Kumar, *Enforcing the GNU GPL*, 1 J. L. TECH. & POL'Y 1, 11 (2006).

[49] Ira Heffan, *'License Compatibility' in Open Source Licenses*, Practising Law Institute, Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, 916 PLI/Pat 11, 16 (2007).

[50] Gonzalez, *supra* note 7, at 173.

[51] *Id.* at 185.

[52] Matthew A. Neco and Wendy Millar Goodkin, *Free and Open Source Software: Risks Your Company's Software Developers Might Not Be Aware Of*, in UNDERSTANDING OPEN SOURCE SOFTWARE, *available at* http://www.acc.com/resource/v6861. *See also* Stern & Kane, *supra* note 1, at 43.

[53] Stern & Kane, *supra* note 1, at 43.

Berkeley Software Distribution (BSD) and other Permissive Licenses

The Open Source Initiative (OSI) sets forth ten requirements for a license to receive recognition for complying with the OSI's principles. They include the right to free distribution, accessibility of underlying source code, possession of the option and capability to revise derivative works, protection of the integrity of the original author's code, no discrimination against individuals or groups, distribution of the license with all subsequent works derived in whole or in part from others' code, prohibition of restrictions on other software, and finally, the license must be "technology neutral."[54] The OSI website provides descriptive details for each of the ten principles.[55]

Some open source licenses permit the free sharing of source code among developers without requiring them to make public and to license forward their innovations and modified versions of the software.[56] These licenses are referred to as permissive or "non-copyleft" licenses.[57] Unlike the GPL licenses, these permissive licenses do not contain copyleft clauses. Examples include the Berkeley Software Distribution (BSD) license, Apache, and others.[58] These licenses differ from hereditary licenses primarily in their treatment of programmers' ability to limit access to the derivative works created from the initial open source code. The BSD license, for example, permits free software to be turned into proprietary software without requiring that the derivative work be redistributed under the original license.

---

[54] *See* source cited *supra* note 15.

[55] *Id*.

[56] Stern, *supra* note 3, at 218.

[57] *Id.*

[58] *Id.* at 214-15.

General Public License Version Three (GPLv3)

Version 3 of the General Public License was released at the end of June 2007.[59] While its release was a monumental event for the software community, it was overshadowed by the release of Apple's iPhone on the very same day.[60] Generally, version 3 reflects the philosophy and values of the open source community.[61] Typically the parties involved when dealing with open source products include the vendors, contractors, and customers.[62] Breaching the GPL becomes most pressing when the software is distributed to third parties and so for obvious reasons, it is important to understand the key elements of the license.[63] In the business world, these parties are typically acquirers (those organizations or companies looking to buy another business entity), targets (those entities being evaluated by acquirers to become part of the acquiring company's organization), and customers (meaning the consumers – either individuals or business – that serve to generate revenue for a business organization). Because of the high risk of subsequent lawsuits and publicity debacles if open source code is improperly handled, many customers, potential partners, and potential acquirers are demanding indemnification clauses at the outset of a deal.[64]

---

[59] Stephen J. Davidson & Nathan S. Kumagai, *Developments in the Open Source Community and the Impact of the Release of GPLv3*, Practising Law Institute: Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, 929 PLI/Pat 319, 325 (2008).

[60] *Id.*

[61] *Id.* at 327.

[62] Stern, *supra* note 3, at 192.

[63] *Id.*

[64] *Id.*

There are five notable revisions in GPLv3 of which corporate attorneys, in-house counsel, and intellectual property lawyers should be aware. First, version 3 permits software distributors to restrict, but not eliminate, the effect patents have on redistribution use of open source programs.[65] This is accomplished in section 11 of the license by explicitly offering a patent grant and then defining it so there is no room for ambiguity when the license is executed.[66] "Patent license" is defined as "any express agreement or commitment…not to enforce a patent."[67] Version 2 was much stricter regarding the intersection of software patents and the GPL.[68] In fact, the preamble states, "we wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all."[69]

The next notable clause addresses "cross-licensing" agreements.[70] These agreements apply to both users who are parties to the agreements and users who are not parties to the

---

[65] From Section 11 of GPLv3: "Each contributor grants . . . a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version." GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html.

[66] *See id.*

[67] *Id.*

[68] GNU General Public License, version 2 (1991), http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt.

[69] *Id.*

[70] Stern From Section 10 of GPLv3: "Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License….If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever

agreements.[71] The language in GPLv3 addressing these agreements essentially states that the licensor and licensee agree not to sue each other for software patent infringement.[72] This condition was absent in version 2.[73] The decision to include this new provision is a direct result of the conflict between Microsoft and Novell in which the former agreed not to enforce its patents against the latter's SuSE Linux customers.[74]

Tivo-ization refers to the practice of employing software licensed under the GPL in a hardware device (such as the popular TiVo digital video recorder used to record television programs) thereby preventing users from sharing the underlying source code.[75] This is a blatant violation of the open source philosophy's freedom to retain, share and allow access to source code. Tivo-ization thus undermines this important goal of the open source movement.[76] The third revision addresses this issue.[77] Version 3 added a clause to ensure open source works under

---

licenses to the work the party's predecessor in interest had or could give[.]" GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html.

[71] *See supra* note 70 and accompanying text.

[72] Davidson & Kumagai, *supra* note 59, at 199.

[73] GNU General Public License, version 2 (1991), http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt.

[74] Microsoft provided a license for its intellectual property directly to the end user rather than the developer (Novell). For details on this controversy, *see* Tom Sanders, *Microsoft Signs Up LG for Linux Patent Deal*, V3, Jun. 8, 2007, http://www.vnunet.com/vnunet/news/2191649/microsoft-signs-lg-linux-patent.

[75] *An Introduction to Tivoization*, The Linux Information Project, http://www.linfo.org/tivoization.html (last visited May 11, 2010).

[76] *Id.*

[77] GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html.

the GPL which will continue to permit future downstream modifications.[78] It is known as the "anti-tivo-ization" clause.[79] Section 1 of version 3 requires licenses to provide "complete and corresponding source" code which includes instructions and information sufficient for a user to execute modified versions of the software. [80] Those most affected by this clause are distributors of devices like TiVo, because the clause does not permit modified code version to run in those distributors' products.[81]

Version 3 goes beyond previous versions of the GPL by not only identifying when covered works must be licensed under the GPL, but by also identifying when derivative works must be licensed.[82] This is a significant modification to the license because despite the self-perpetuating nature of the GPL, the change articulates circumstances in which a work derived under a hereditary license does not necessarily self-perpetuate in a manner detrimental to the copyright of works aggregated with it.[83] [84]

---

[78] GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html.

[79] Stern, *supra* note 3, at 223.

[80] "If you convey an object code work under this section in, or specifically for use in, a [consumer] Product . . . the Corresponding Source conveyed . . . must be accompanied by the Installation Information . . . Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documents (and with implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying." GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html (referring to § 6 specifically).

[81] Stern, *supra* note 3, at 223.

[82] Stern & Kane, *supra* note 1, at 44.

[83] Heffan, *supra* note 49, at 16.

[84] "A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an

Finally, the definition of "corresponding source" is included in the text of the license so that licensees are aware that incorporated and/or copied code is defined to include code used to modify, install and run programs.[85] Version 2 addressed corresponding source code in its section 3(b), but included no description or definition of it, thus causing confusion for users.[86]

The new version of the GPL has not been well received by the entire open source community. Linus Torvalds, a well-known GPL supporter and open source enthusiast, has made his acceptance of version 3 contingent upon Sun Microsystems' release of Solaris under GPLv3.[87] He cites as his concerns interference with hardware and software, and he questions the appropriateness of some revisions. On the other hand, GPLv3 has received strong support from the Samba Project, Free Software Foundation, IBM and Sun Microsystems. How it will ultimately be received by the open source community remains to be seen.

'aggregate' if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate." GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html (referring to § 5 specifically).

[85] "The 'Corresponding Source' for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities . . . The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source. The Corresponding Source for a work in source code form is that same work." GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html (referring to § 1 specifically).

[86] GNU General Public License, version 2 (1991), http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt.

[87] Davidson & Kumagai, *supra* note 59, at 142.

Compatibility issues between licenses

With the variety of open source code licenses available, compatibility problems have arisen.[88] These complications are more related to the legal provisions in the licenses, more so than technical compatibility issues.[89] Because of these difficulties the ways in which code from multiple sources may be properly combined under the terms of the licenses are limited.[90] Perhaps the most remarkable consequence of the incompatibility is that two sets of source code licensed under reciprocal, or hereditary, licenses could not be combined under another reciprocal license.[91] This is because each license would require the resulting work be covered under a license identical to itself.[92] Code covered under the GPL is included in this licensing conundrum. To be compatible with the GPL, the license must contain a clause embodying the spirit of the copyleft clause from the original GPL.[93] For example, if a GPL product were combined with a product covered under the Artistic License (another copyleft, or hereditary, open source license), each license would require that the derivative program be licensed under its own terms. This is a complicated and grave legal problem if the code under these respective licenses became co-mingled post-merger or acquisition. Fortunately, the new GPLv3 resolves

---

[88] *Open Source Licenses By Category*, Open Source Initiative, http://www.opensource.org/licenses/category. Click on any link for the various categories of licenses and it becomes readily apparent that there is great variety in conditions and requirements for each category of open source license.

[89] *Id.*

[90] Stern & Kane, *supra* note 1, at 44.

[91] *See* GNU General Public License, version 2 (1991), http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt; *see also Artistic License 2.0*, Open Source Initiative, http://www.opensource.org/licenses/artistic-license-2.0.php.

[92] *See* sources cited *supra* note 91.

[93] *Id.*

this quandary by providing that open source covered under it or a similar reciprocal (hereditary) license may be combined with code covered by a non-reciprocal license.[94]

Another example of incompatibility between licenses arises where a derivative work contains code covered by a permissive license such as the BSD and also contains code covered by the GPL. The BSD terms are more permissive than those of the GPL, meaning the license does not require a derivative work to be licensed under the BSD itself. This is one of the reasons it is popular among commercial software developers. However, when work covered under a license similar to the BSD is combined or commingled with work covered under a GPL version 2, the work must be subsequently licensed under the GPL, and not the BSD.[95] The GPL's requirements that a derivative work be covered under the same license as the original work would be violated if the BSD license were used instead.

From a corporate transactions perspective, not only must attorneys determine whether open source code has been used in a target company's software, but must also determine under what license the code was licensed to the target company. Additionally, it is crucial that legal professionals representing corporations or intellectual property based companies ascertain whether that code is going to be combined with any existing software of a potential acquiring company, and whether the acquiring company's software may be combined with the target company's code.[96] In short, it is critical that legal professionals consider the possibility of incompatible licenses if open source code is to be combined with other code in existing products and licensed further.

---

[94] GNU General Public License, version 3 (2007), http://www.gnu.org/copyleft/gpl.html.

[95] *See* sources cited *supra* note 91.

[96] Stern & Kane, *supra* note 1, at 45.

Acquisitions and the Corporate Context

Products incorporating open source code are proliferating exponentially whether the software developers are aware of its presence or not.[97] Many businesses operate in a "mixed-IP environment." These environments contain many types of programs licensed under many disparate licenses, each with specific provisions, conditions, and prohibitions. As a result, it is becoming more common for new software to be created by using as its building blocks prior pieces of work.[98] Layering of code creates a melting pot of software owned by the developer with code owned by someone else, some of which is likely to be open source.[99] It is unsafe to assume that corporate software engineers are aware of the implications of open source licenses, particularly the reciprocal nature of the commonly used GPL.[100] Since decisions to include open source code in proprietary software are not necessarily made by chief engineers or product managers, management may not be aware of the degree to which open source code is being incorporated into software in most companies.[101] It is critical to ascertain the extent of the use of open source code in any company.[102]

---

[97] Stern, *supra* note 3, at 191.

[98] Kat McCabe, *Working with Open Source: Software Compliance Management*, Practising Law Institute: Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, 927 PLI/Pat 473, 477 (2008).

[99] *Id.*

[100] Stern, *supra* note 3, at 191.

[101] *Id.*

[102] McCabe, *supra* note 98, at 477.

Knowing the value of a target's intangible software assets or of its utility for a given purpose is insufficient for proper conduction of a deal valuation.[103] The target's rights to its intellectual property must be ascertained and, quite likely, this will entail tracing the software's development from a legal perspective.[104] It is not difficult to foresee a scenario where, if a target's intellectual property assets are inaccurately valued, the perceived value of the deal will likely be inaccurate as well.[105] The less exclusive the rights a company has to its intellectual property assets, the less that company's value is.[106] The results of a miscalculated valuation of a corporate transaction may include unanticipated competition from other rightful owners and users of the assets, or, if the assets that were once proprietary have lost their exclusivity due to mismanagement of licensing rights, there may be no value to the intellectual property assets at all.[107] Clearly, these results are undesirable but given the proper training and information, savvy legal professionals will be well-equipped to manage open source licensing issues.

Valuation of a prospective acquisition becomes more critical when a sizeable percentage of a company's assets are composed of intellectual property.[108] Potential purchasers should identify the scope and value of a target's intellectual property assets so as not to over-pay for the

---

[103] Anthony F. Lo Cicero et al., *Acquiring or Selling the Privately Held Company 2008*, 1675 PLI/Corp 733, 739 (2008).

[104] *Id.*

[105] *Id.*

[106] *Id.*

[107] *Id.*

[108] Lo Cicero, *supra* note 103, at 737.

acquisition, and in order to maximize the value of those assets after the transaction.[109] Licensed intellectual property rights are not necessarily transferable by the licensee without express permission by the licensor, thus decreasing the value of the licensee's rights to the asset.[110] The desirability of the target of an acquisition may or may not stem largely from its intellectual property. Either way, mismanagement of open source code can derail a deal or devalue a company very quickly. If a target's cornerstone products serve as the main attraction and impetus for the deal, and those products were developed by software engineers through the use of open source code, it does not take much of a leap to see how quickly the company could be devalued in the eyes of the potential purchaser, and the deal itself placed in jeopardy.[111] The risk of lawsuits and the potential for unfavorable publicity may also discourage other companies considering entering into a bidding war with the known suitor for the target and so market competition and efficiency are diminished.[112] For obvious reasons, due diligence with respect to assessing intellectual property assets has become a prominent concern and at the forefront of the minds of savvy corporate, in-house, and intellectual property attorneys.[113]

Some executives or product managers might prefer the "ignorance is bliss" approach when it comes to licensing structures.[114] Quite frankly, it is often less expensive for businesses to build new software based on previous works, whether those works are for internal use at the

[109] Lo Cicero, *supra* note 103, at 739.

[110] *Id.* at 737.

[111] Stern, *supra* note 3, at 194.

[112] *Id.*

[113] *Id.*

[114] Stern & Kane, *supra* note 1, at 46.

company that employs the developers, or for external use whereby the company will pass the work to outside parties for profit.[115] Open source is almost always seen as a less expensive alternative to proprietary assets.[116] Time and development costs are reduced while quality assurance and reliability are potentially increased.[117]

It is tempting for upper management to address problems that appear simple to resolve, that are quickly fixable and manageable, rather than those that are tedious, difficult and less likely to be resolved with ease.[118] It is simply part of human nature to first deal with what is easily and readily fixable. Corporate entities have an enormous interest in identifying whether open source software may be present and if so, how it is being used and managed. Thus, extensive due diligence should be conducted prior to the acquisition of another company's intellectual property assets.[119] When forming strategic alliances, potential partners are currently demanding higher levels of assurance regarding intellectual property assets.[120] Investors and customers do not wish to be unpleasantly surprised to learn that a business may be vulnerable to lawsuits or loss of competitive advantage because of mismanagement of open source code.[121] This holds true for employees as well.[122]

---

[115] McCabe, *supra* note 98, at 477.

[116] Lo Cicero, *supra* note 103, at 755.

[117] McCabe, *supra* note 98, at 477.

[118] *Id.*

[119] *Id.*

[120] *Id.*

[121] *Id.*

[122] McCabe, *supra* note 98, at 477.

Software Compliance Management and Due Diligence

Open source code carries with it serious risks that must be addressed before reaping the benefits of employing such software. It is remarkably easy to "license in" a publicity or legal nightmare.[123] In the corporate context and particularly where an acquisition is involved, it is necessary to understand that the identification of open source and compliance with license requirements is absolutely critical to the success of the deal and the survival of the parties involved.[124]

So where should a company begin? The appropriate standard of care is for the business to conduct due diligence. This entails a proper assessment of what a target is worth, what the value of its liabilities may be, will likely require a pre-acquisition audit of intellectual property assets.[125] Companies should begin by assessing the assets they are acquiring.[126] There are numerous rationales behind this including making sure the internal review is not unnecessarily complicated by the procedures used. A great deal of learning is likely to occur during the review process and it is easier to educate employees as opposed to third parties.[127] That is probably due to the simple fact that employees are on the premises and are easier to reach than outside parties. The goals of accurate asset valuation and identification are shared by all businesses that find themselves going through the major undertaking of a merger or acquisition.[128]

---

[123] Stern, *supra* note 3, at 192.

[124] Stern & Kane, *supra* note 1, at 42.

[125] Lo Cicero, *supra* note 103, at 738.

[126] McCabe, *supra* note 98, at 479.

[127] *Id.*

[128] *Id.* at 478.

Due diligence conducted when open source is at issue is not necessarily different from due diligence conducted for any other reason.[129] When managing open source both internally and during the acquisition of a target, a company's goals should include maintenance of continuity in its business model and compliance with legal obligations.[130] Checklists, drafts of representations and warranties, oral interviews all facilitate the process of due diligence.[131] External forensic computer consultants who specialize in programming may be hired to peel back the code layer by layer to determine whether open source might be present.[132] For example, Black Duck and Palamida are two companies that specialize in and provide such services in this arena.[133] Internal engineers can also be used to scrutinize code for any potential signs or indicators of open source in the assets.[134]

It is not hard to understand why a new field known as Software Compliance Management has emerged.[135] Its purpose is to offer guidance and facilitate transactions and manage internal assets.[136] Generally, Software Compliance Management can be understood to mean personnel or divisions that perform the function of, or the process of:

---

[129] McCabe, *supra* note 98, at 479.

[130] *Id.* at 478.

[131] *Id.* at 479.

[132] *Id.*

[133] Stephen J. Davidson & Nathan Kumagai, *Developments in the Open Source Community and the Impact of the Release of GPLv3*, Practising Law Institute: Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, 916 PLI/Pat 121, 141 (2007); *see also generally* http://www.palamida.com and http://www.blackducksoftware.com.

[134] McCabe, *supra* note 98, at 479.

[135] *Id.* at 478.

- "Knowing what is in the code base and controlling the introduction of licensed materials into the code base;
- Knowing what obligations are related to the use of licensed materials, knowing whether the license obligations for combined components are compatible, and managing fulfillment of those obligations; and
- Managing this process from the product architecture all the way through deployment or distribution."[137]

Software Compliance Management provides assurance that the company's objectives are achieved and licensing obligations are complied with when using or incorporating new intellectual property assets.[138] An important consideration is whether the software licenses are assignable.[139] The purchaser should determine whether the target has received requests from rights owners for licenses and for the payment of licensing.[140] The Sarbanes-Oxley Act of 2002 was enacted as a corporate governance measure and is relevant to a discussion of performing due diligence.[141] Sarbanes-Oxley includes intellectual property assets under the company assets that must be monitored regularly.[142] Any existing impairments to a company's intellectual property assets must be disclosed.[143] If open source code is used improperly it can significantly damage an intellectual property asset. Sarbanes-Oxley compliance attorneys should communicate openly

---

[136] McCabe, *supra* note 98, at 478.

[137] *Id.*

[138] *Id.*

[139] Lo Cicero, *supra* note 103, at 754-55.

[140] Stern, *supra* note 3, at 194.

[141] *See* Sarbanes-Oxley Act of 2002 § 7202, Pub. L. No. 107-204, 116 Stat. 745 (2002).

[142] *Id.*

[143] *Id.*

with corporate finance departments to develop auditing procedures and ensure proper drafting of the certification.[144]

Certain problems are commonly faced by acquiring companies when open source software is involved. First, the improvements made to the open source code may be required to be publicized to the open source community depending on the requirements of the applicable licenses. Also, combining proprietary code with open source code could potentially render all the company's commercial and proprietary products subject to an open source license. Companies fearful of infecting their proprietary software with hereditary-licensed open source code may have the option of distributing the derivative work under a fee-based commercial license.[145]

Remediation

Despite all we know about open source and its exponential growth we have witnessed, there remain unanswered questions as to how to best track it and achieve compliance with relevant business standards and laws. Such questions include ascertaining the factors needed to conclude that copied work is *de minimus*, when do functions and expressions merge to the degree that we are no longer examining copyrightable code, and what is the responsible depth of investigation before accepting a developer's representation that the code was not wrongfully copied?[146]

Remediation is the next step in the open source management project following the performance of due diligence and can be considered the most challenging stage of the

[144] Stern & Kane, *supra* note 1, at 46.

[145] Stern, *supra* note 3, at 196.

[146] McCabe, *supra* note 98, at 480.

assessment.[147] Remediation refers to the process by which an entity can correct prior mistakes and perform damage control if necessary. Attorneys without extensive technical knowledge or background would be remiss if they did not consult with outside counsel who possesses the expertise to communicate directly with developers regarding remediation options and procedures.[148] The professionals consulted should be familiar with open source, cognizant of its development, know where to retrieve valuable information if needed, and ideally, have personal contacts in the open source community that could be consulted further.[149]

It is critical that there be open channels of communication among the attorneys involved in remediation, the management and the developers.[150] Sometimes companies will consent to specifics of a remediation plan of action and even secure their positions using escrow or holdbacks.[151] Failures or problems revealed by due diligence reviews vary in degree of seriousness and remediation procedures should be adjusted according to the severity of the problem.[152] Remediation costs and procedures may present formidable, challenging decisions for the company.[153] If compliance management inspection revealed that remediation can be achieved through replacement of noncritical code, many companies remain comfortable in

[147] McCabe, *supra* note 98, at 480.

[148] *Id.* at 484.

[149] *Id.*

[150] *Id.*

[151] *Id.* at 480.

[152] McCabe, *supra* note 98, at 480-81.

[153] *Id.* at 480.

proceeding with the transaction.[154] Often the remediation efforts will be left to the acquiring entity.[155] Unfortunately there are cases where inspections have uncovered serious circumstances of non-compliance.[156] If efforts to meet the necessary compliance standards failed, it can become challenging compromise on which entity (the target or the acquirer) will bear the costs of remediation.[157] It is likely the deal will be delayed as a result of having to conduct remediation procedures.[158] Where a deal is on the line and problems are identified, the best case scenario for complex remediation procedures is that the transaction will be delayed.[159] The worst case scenario would be for the acquirer to lower the price they are willing to pay or to cancel the deal in its entirety.[160] It is preferable for a company to voluntarily perform a due diligence review prior to having one forced upon it – substantial value and opportunity can potentially be lost if the latter occurs.[161]

The degree of sensitivity an acquiring company will have towards open source code depends in large part on the intended use of the software.[162] If the acquisition's purpose is to temporarily fill a hole in the acquiring company's product line, and the target's assets are

---

[154] McCabe, *supra* note 98, at 480.

[155] *Id.*

[156] *Id.*

[157] *Id.*

[158] *Id.*

[159] McCabe, *supra* note 98, at 480.

[160] *Id.*

[161] *Id.*

[162] *Id.* at 481.

intended to remain as part of the subsidiary and not be rolled up into the parent, there may be relatively little sensitivity to problems that might be disclosed through extensive due diligence. The intent, or end-goal, of the corporate transaction may be for the acquired code to be replaced in its entirety by code developed as an integral part of the company's existing product line. This is not an ideal valuation scenario for the target company, but it will result in less need for code scans or other methods of detailed due diligence review. Whatever the acquiring company's motivation, if the intention is to merge the acquired code into the acquiring company's existing product line and combine it with other valuable software assets, then the sensitivity will be much higher and the discovery of even minor problems may be sufficient to tip the balance toward 'make' in a 'make versus buy' analysis.[163] During a merger or acquisition, depending on what the goals of the acquiring company are, if the acquiring company intends to replace its target's software with whatever the acquirer has developed internally, the acquirer will be more likely to continue to manufacture its software in-house and not pay additional cost, or pay a lower cost to acquire the intellectual property assets of the target. Contrarily, if the acquiring company seeks to replace or commingle its programs with those of the target, the price of the deal may increase because of the associated due diligence and potential remediation costs.

Enforceability of Open Source Licenses and Litigation

Traditionally, the failure to adequately investigate the use of open source code has not been in the forefront of legal professionals' concerns.[164] Until recently, there were no developers in a position to instigate any type of infringement action.[165] This relaxed attitude toward

[163] McCabe, *supra* note 98, at 481.

[164] Stern & Kane, *supra* note 1, at 46.

[165] *Id.*

compliance with open source licenses is likely to change in light of recent developments in litigation centering around the enforceability of open source licenses.[166]

Most strikingly, the U.S. Court of Appeals for the Federal Circuit recently found in favor of a copyright holder in the case of *Jacobsen v. Katzer*.[167] The outcome of this action addressing open source license enforceability was the reversal of the lower court's denial of a preliminary injunction. In *Jacobsen* the court held that plaintiff, who had created software used in conjunction with model trains and which was covered under the Artistic License (a hereditary, or viral, license similar to the GPL), was entitled to make a claim not only for violation of the license, but also for copyright infringement. Additionally he was granted the option to enjoin the defendants who copied the licensed software without adhering to the conditions set forth in the Artistic license. Certain features of the license, such as those ensuring access and the ability to modify code, as well as the requirement to make attribution to the author, were found to establish conditions limiting the scope of the license granted to the defendant.[168] They were not merely covenants and these features of the license were found to be violated by the defendants. By violating the license terms, the defendants exceeded the scope of the license; their conduct gave rise to an action for copyright infringement rather than merely for breach of contract.[169] While this opinion is encouraging for supporters of the open source movement, in-house counsel should pay close attention since it appears that violators of open source licenses could be exposed to copyright infringement liability.

---

[166] Stern & Kane, *supra* note 1, at 46.

[167] *Jacobsen*, 609 F. Supp.2d at 932.

[168] *Id*.

[169] *Id.* at 933.

In 2007 and 2008, the Software Freedom Law Center (SFLC) initiated lawsuits on behalf of Erik Andersen and Rob Landley, the principal developers of BusyBox.[170] The SFLC is the pro bono branch of Stallman's Free Software Foundation. [171] These lawsuits claimed that Verizon, Monsoon Multimedia, High-Gain Antennas and Xterasys, Super Micro Computer, Inc., and Extreme Networks, Inc. violated version 2 of the GNU General Public License (GPLv2). The grounds for the claims were alleged distribution by the defendants of products that contained BusyBox code. The critical factor was that these products were released without the underlying source code, thereby breaching the terms of the GPLv2.[172] As of October 2008, only Extreme Networks action remained outstanding.[173] Generally the settled actions contained two similar components in their resulting agreements. First, the defendants agreed to appoint an internal Open Source Compliance Officer who would function to monitor compliance with GPL licenses, and to publish the source code for the version of BusyBox they previously distributed.[174] The second element of the settlements involved undisclosed payment by defendants to BusyBox's developers.[175]

There are important points to take away from the recent settlements and litigation. The redistribution of software under hereditary licenses is complex. It can easily frustrate even a

---

[170] The complaints were all filed by the Software Freedom Law Center, Inc. For the complaint against Monsoon Multimedia, Inc., *i.e.*, the first suit filed, see *Anderson v. Monsoon Multimedia, Inc*. Complaint No. 07-cv-08205-JES (S.D.N.Y. Sept. 19, 2007), *available at* http://www.softwarefreedom.org/news/2007/sep/20/busybox/complaint.pdf.

[171] Stern & Kane, *supra* note 1, at 45.

[172] *See* source cited supra 170, at 5.

[173] Stern & Kane, *supra* note 1, at 45.

[174] *Id.*

[175] *Id.*

savvy, sophisticated company's ability to manage its exposure to risk. Consequently, focus on internal compliance is extremely important. Companies using open source code cannot rely on the ill-conceived notion that open source licensing is irrelevant to their line of work.[176]

Moreover, the SFLC created Moglen Ravicher LLC, a firm employing the very lawyers who instituted the original five suits discussed above, to represent for-profit clients in GPL license violation claims.[177] It is foreseeable that, going forward, more firms will undertake this type of litigation on a contingency fee basis.[178] Actions regarding open source licensing enforceability are not limited to the United States.[179] In 2007, Harald Welte brought a series of German enforcement actions for alleged violations of the GPLv2.[180] Welte was in the business of running the gpl-violations.org project. The court found in his favor in one such suit against Skype Technologies SA.[181]

Open source software invites a plethora of legal interpretations and applications based just on the few actions that have been litigated.[182] For example, courts have been faced with the question of determining whether a license agreement was created.[183] Typically programmers using open source code do not sign a license agreement and the threshold issue in some actions is

---

[176] Stern & Kane, *supra* note 1, at 46.

[177] *Id.*

[178] *Id.*

[179] *See generally* Harald Welte's Blog, http://laforge.gnumonks.org/weblog/linux/gpl-violations/.

[180] *Id.*

[181] *Id.*

[182] Stern & Kane, *supra* note 1, at 46.

[183] *Id.*

degree of enforceability of "browsewrap" agreements.[184] The Free Software Foundation and other attorneys have taken the view that open source licenses are not contracts.[185]

Another area of dispute centers around whether the nature of open source software creates infringement issues where there are existing patents. Microsoft has publicly alleged that the Linux operating system (which competes against Microsoft's Windows) violates a large number of Microsoft's patents and that other open source code infringes upon over 200 Microsoft patents.[186] Despite the various interpretations of legal issues surrounding open source code, the risks associated with its use in commercial settings remain high as a result of the lack of guidance from the courts in this developing area.

CONCLUSION

Despite its risks, open source carries with it many benefits. It can be a cost-effective alternative to purchasing software. The communal contributions and collaboration of ideas that sculpt and create the derivative, modified or redistributed works bring highly positive synergistic results that cannot be denied. With the right group of informed professionals, open source can be managed effectively.[187] Open source code and the associated licenses may seem daunting at first, particularly to lawyers accustomed to having a tangible, signed paper agreement before use

---

[184] LORI LESSER, *Open Source Software 2006: Critical Issues in Today's Corporate Environment*, Practising Law Institute: Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series, 885 PLI/Pat 9, 26 (2006).

[185] *See* Matt Lee, *What is Free Software and Why is it so Important for Society?*, FREE SOFTWARE FOUNDATION, March 29, 2010, http://www.fsf.org/about/what-is-free-software.

[186] Roger Parloff, *Microsoft Takes On The Free World*, FORTUNE, May 14, 2007, http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/. Microsoft has also entered into a number of "cross-licensing agreements," which may really be cross-covenants not to sue with open source software companies regarding IBM's patents.

[187] Stern & Kane, *supra* note 1, at 49.

of the software can begin; however, open source code can be manageable with the cooperation of software development teams and other legal and compliance professionals.

Attorneys have a vested interest in broadening their horizons of knowledge on open source software. The omnipresence of the software makes it critical to the legal profession, particularly for general counsel and even more critical for in-house counsel to technology and engineering firms. Awareness, understanding and a basic plan for due diligence is crucial for lawyers who will absolutely be faced with the ramifications of the GPLv3 on proprietary intellectual property assets. With pro-activity and open-communication channels between upper management, legal counsel and engineers, open source can be managed effectively and safely. The ease and benefits of its use have contributed to its prevalence, requiring the attention of internal counsel, corporate attorneys and intellectual property attorneys.