



WAKE FOREST
INTELLECTUAL PROPERTY
LAW JOURNAL

VOLUME 10

2009 - 2010

NUMBER 3

NAVIGATING THE OPEN SOURCE MINEFIELD: WHAT'S A BUSINESS
TO DO?

T. Robert Rehm, Jr.¹

It goes without saying that computers and the software programs² that run on them permeate virtually every aspect of our professional and personal lives. For many years, software developers licensed³ their software products pursuant to a regime designed to vigorously protect the developers' proprietary intellectual property rights in those products. Licensing software only in object code form and contractually prohibiting access to and use of the source code of

¹ Smith, Anderson, Blount, Dorsett, Mitchell & Jernigan, L.L.P., Raleigh, North Carolina. B.S. Engineering Science (*summa cum laude*), University Of Notre Dame (1971); M.B.A., University of Alabama-Birmingham (1990); J.D. (*magna cum laude*), Cumberland School of Law of Samford University (1993).

² A "computer program" is a "sequence of instructions written to perform a specified task for a computer," RALPH M. STAIR ET AL., PRINCIPLES OF INFORMATION SYSTEMS 132 (2nd ed., Thomson Learning, Inc. 2003), while "computer software" is a computer program stored in digital format. "Computer software", *available at* <http://dictionary.reference.com/browse/computer%20program> (last visited May 25, 2010). Because the manner in which source code is stored is not germane to the discussion herein of source code as copyrightable subject matter, the terms "computer software," "software," "computer program," and "program" will be used interchangeably in this article.

³ Ownership of copyright in computer software provides certain exclusive rights to the owner in connection with that software. *See* discussion *supra* Parts I.B. and I.C. and accompanying footnotes. Within a software "license" agreement, the owner of copyright in software (i.e., the "licensor") typically relinquishes, via a grant of license, some or all of the licensor's exclusive copyright rights to another party (i.e., the "licensee"), subject to certain limitations and restrictions on the licensee's exercise of such rights. H. WARD CLASSEN, A PRACTICAL GUIDE TO SOFTWARE LICENSING FOR LICENSEES AND LICENSORS 11, 18-19, 21-38 (3d ed. 2008) [hereinafter CLASSEN ON LICENSING]. Typically, the scope of a license grant to use software is broad enough to permit the licensee to use the software in the manner intended by the parties but narrow enough to enable the licensor to rely on copyright law and the terms on the license agreement to prevent unauthorized, infringing uses of the software by the licensee and others. *Id.* at 18.

such products – *i.e.*, a “closed” source licensing approach – have been and continue to be hallmarks of that regime.⁴

In recent years some software developers have adopted a different approach to licensing software, which permits licensees to not only access the software source code but also reproduce, distribute, and even modify the source code with a view to providing more opportunities for enhancing the software’s capabilities and correcting its shortcomings. This new approach has caused a good bit of confusion, and resulting consternation, among organizations that rely in any way on software in conducting their businesses. The most often asked questions are: Does (or should) the organization use any open source software in its business? If the organization does use (or plans to use) such software, what steps, if any, should be adopted and implemented to maximize the benefit, while managing the risk, associated with such software?

These rather open-ended questions hopefully will come into sharper focus by the time we reach Part III of this article, at which time we will address them head-on. With that objective in mind, this article sets out to (1) explain, from both legal and business perspectives, the traditional (*i.e.*, “closed” source) model of licensing computer software and the “open” source licensing model, (2) identify the common ground and highlight some of the more important differences in the two models, (3) comment on the ongoing viability of the open source model as a licensing methodology, and (4) provide

⁴ CLASSEN ON LICENSING, *supra* note 3, at 34-35 (“[M]ost licensors refuse to sell source code licenses[,]” which provide licensees access to the source code of the licensed software, because such licenses (1) undermine the objectives of enforcing trade secret protection for source code and, thereby, increase the risk of its misappropriation by licensees; (2) make it easier for licensees to create unauthorized derivative works of the licensed software; and (3) increase the likelihood that the licensee will use the licensed source code to compete with the licensor or will disclose the licensed source code to a competitor of the licensor.). Such unauthorized derivative works presumably go into the public domain by operation of Section 103(a) of the Copyright Act. *See infra* note 36. In order to maintain (*i.e.*, correct bugs and other errors) and support (*i.e.*, update, upgrade, and enhance) computer software, it is necessary to access and modify the software source code. CLASSEN ON LICENSING, *supra* note 3, at 34. Therefore, when a software licensor is no longer willing or able to maintain and support licensed software (e.g., due to the licensor’s bankruptcy or insolvency or refusal to provide maintenance and support services for any reason), even a software licensee under a closed source licensing model needs access to software source code so that it can provide such services itself or through a third-party vendor. *Id.* at 181. Such access is typically provided via a source code escrow arrangement, wherein a source code escrow agent is responsible for releasing the source code of the licensed software to the licensee upon the occurrence of certain events, such as those described in the preceding sentence. *Id.* at 181-84.

information that a business can use to assess whether open source software meshes well with the business's goals and objectives.

I. Just How Different Are the Closed Source and Open Source Licensing Models?

As a threshold matter, it is important to note at the outset that the fundamental principles giving rise to the differences between the closed source and open source models for licensing software are rooted neither in the software itself nor in the intellectual property protection available for such software⁵ but, rather, in the manner in which that software is licensed or otherwise distributed to users. Therefore, in order to narrow the scope of issues that businesses ultimately must address in deciding whether and how to utilize open source software, it is helpful to identify and describe the following

⁵ Computer software is subject to protection under copyright law as a work of authorship and also may be subject to protection under patent law to the extent of any patentable business methods included therein or under trade secret law to the extent of any information embodied in the software that derives independent economic value from being maintained in secret and is subject to efforts reasonable under the circumstances to maintain such secrecy. CLASSEN ON LICENSING, *supra* note 3, at 12-16. Patent protection of software-based business methods first came on the scene in 1998 via the decision of the United States Court of Appeal for the Federal Circuit ("CAFC") in *State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998), which "open[ed] the floodgates on software patents." CLASSEN ON LICENSING, *supra* note 3, at 11-12. The ongoing viability of patent protection for software-based business methods rooted in the "useful, concrete, and tangible" result test espoused in *State Street Bank* was thought to be in imminent jeopardy, because, in its decision in *In re Bilski*, 545 F.3d 943 (Fed. Cir. 2008) (en banc), *cert. granted*, 129 S. Ct. 2735 (2009), the CAFC states that "the 'useful, concrete, and tangible result' inquiry is inadequate" and "those portions of our opinion[] in *State Street [Bank]* . . . relying solely on a 'useful, concrete, and tangible result' analysis should no longer be relied on." *Bilski* at 960, n.19. The Supreme Court rejected the new "machine-or-transformation" test for process patentability articulated by the CAFC in *Bilski* as the "sole test for deciding whether an invention is a patent-eligible 'process'" and characterized that test as "a useful and important clue, an investigative tool, for determining whether some claimed inventions are processes under § 101." *Bilski v. Kappos*, No. 08-964, slip op. at 8 (U.S. June 28, 2010). Due to the somewhat uncertain status of patent protection for software, this article does not discuss such protection as it relates to the closed source and open source licensing models. Pursuing protection for software under trade secret law is a potentially risky proposition, because such protection immediately may be lost upon any intentional or inadvertent disclosure of the applicable trade secret information (e.g., by reverse engineering software to ascertain its source code). CLASSEN ON LICENSING, *supra* note 3, at 13. Because none of the differences in the closed source and open source licensing models appears to have a basis in trade secret law, this article does not further address trade secret protection for computer software.

items, all of which apply with equal force and in identical fashion to both the closed source and open source licensing models: (1) copyrightable subject matter included in computer software; (2) how ownership of copyright rights in such subject matter initially vests and some of the principles governing transfers of such rights; and (3) rights reserved exclusively to the owner of copyright rights in computer software. An understanding of and appreciation for these concepts may help to reinforce the notion that, when taking account of all relevant aspects of a software licensing model – *i.e.*, the software, its applicable intellectual property protections and rights arising out of such protections -- the closed source and open source licensing models actually have quite a bit in common.

A. Copyrightable Subject Matter in Computer Software.

Computer software source code comprises the set of human-readable instructions developed by a computer programmer to specify actions that the computer must take to achieve the objectives of the program.⁶ Computer software object code, on the other hand, is source code that has been compiled into a machine-readable format for execution by a computer.⁷

The U.S. Copyright Act of 1976⁸ (“Copyright Act”) provides protection for “original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include . . . literary works”⁹ In addition, the Copyright Act defines a “computer program” as “a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.”¹⁰

Although the Copyright Act does not expressly provide that computer programs constitute original works of authorship susceptible to protection under the Act, it is well established that computer

⁶ LAWRENCE ROSEN, OPEN SOURCE LICENSING – SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW 2 (2004) [hereinafter, ROSEN ON OPEN SOURCE]; CLASSEN ON LICENSING, *supra* note 3, at 34.

⁷ CLASSEN ON LICENSING, *supra* note 3, at 34.

⁸ 17 U.S.C. §§ 101-1332 (2010).

⁹ *Id.* § 102. “Literary works” are “works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.” *Id.* § 101.

¹⁰ *Id.* § 101. The definition of “computer program” was added to the Copyright Act via an amendment to that Act in 1980. *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1247 (3d Cir. 1983) (citing relevant legislative history).

programs do indeed constitute copyrightable subject matter in the form of literary works.¹¹ Both the source code and the object code underlying a computer program are protectable under U.S. copyright law.¹²

B. Ownership of Copyright Rights in Computer Software.

Ownership of copyright rights in a work protected under the Copyright Act initially vests in the author (or authors) of the work.¹³ Such vesting and protection¹⁴ occur immediately upon fixation of the work in a tangible medium of expression,¹⁵ and such copyright rights are separate and distinct from rights in the material object in which the copyrighted work is embodied.¹⁶ Generally, the “author” of any work that is subject to copyright protection is the work’s creator – *i.e.*, the person, who translates the idea underlying the work into a fixed, tangible expression embodying the work.¹⁷ A “work made for hire”¹⁸

¹¹ *Apple Computer*, 714 F.2d at 1247 (citing legislative history of the 1976 Copyright Act, which states that the term “literary works” as used in that statute “includes . . . computer programs”); *Williams Elecs., Inc. v. Arctic Int’l, Inc.*, 685 F.2d 870, 875-77, nn.4-8 (3d Cir. 1982) (holding that “the copyrightability of computer programs is firmly established after the 1980 amendment to the Copyright Act [of 1976]”).

¹² *Apple Computer*, 714 F.2d at 1249 (stating that “a computer program, whether in object code or source code, is a ‘literary work’ and is protected from unauthorized copying, whether in its object or source code version”); ROSEN ON OPEN SOURCE, *supra* note 6, at 20 (“The source code that defines a computer program is copyrightable, as is the translated object code that actually executes on the computer.”). The expressive elements of computer databases also constitute copyrightable subject matter. 1 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 2.04[C][1], at 2-49 & 2-50 (80th rel. 2009) (citing relevant legislative history, which states that the definition of “literary works” in the Copyright Act “includes computer data bases”). Because the copyrightable aspects of computer databases are not germane to the ultimate inquiry here, this article does not otherwise address the copyright-related or other aspects of such databases.

¹³ 17 U.S.C. § 201 (2010).

¹⁴ Although registration of copyright rights in a work is not required in order for the work to be protected under the Copyright Act, 17 U.S.C. § 408 (2010) (“[R]egistration is not a condition of copyright protection.”), such registration is a prerequisite to instituting, and seeking statutory damages and attorney fees in connection with, any action for infringement of such copyright rights, *id.* §§ 411(a), 412; CLASSEN ON LICENSING, *supra* note 3, at 33, and under certain circumstances constitutes prima facie evidence in any such action of the validity of such rights and of the facts stated in the corresponding registration certificate. 17 U.S.C. § 412 (2010); CLASSEN ON LICENSING, *supra* note 3, at 33.

¹⁵ ROSEN ON OPEN SOURCE, *supra* note 6, at 17-18.

¹⁶ 17 U.S.C. § 202 (2010).

¹⁷ *Cmt’y. for Creative Non-Violence v. Reid*, 490 U.S. 730, 737 (1989) (citing 17 U.S.C. § 102) [hereinafter *Reid*].

¹⁸ A “work made for hire” under the Copyright Act comes in the following forms:

is an exception to this general rule in that, by operation of law and absent a written agreement to the contrary, the author of such a work for copyright purposes is the employer or other party, as the case may be, for whom the work is created.¹⁹

For example, absent a written agreement to the contrary, a computer programmer, who, in other than a “work made for hire” context,²⁰ writes down software source code on a piece of paper owned by another, is the author of, and therefore owner of copyright rights in, such source code. However, the programmer does not take title to the paper merely because the source code is written on it.²¹ The same holds true for computer object code compiled from source code and fixed in a computer diskette – ownership of copyright rights in such object code is separate and distinct from ownership rights in the diskette on which that code is fixed.²² Thus, absent a written agreement to the contrary, transfer of ownership rights in the material object in which (or on which) copyrightable subject matter, such as computer source code or object code, is fixed, does not, in and of itself, constitute a transfer of copyright rights in such subject matter.²³

(1) a work created by an employee within the scope of his or her employment or (2) certain types of specially ordered or commissioned works. 17 U.S.C. § 101 (2010).

¹⁹ *Reid*, 490 U.S. at 737 (citing 17 U.S.C. § 201(b)).

²⁰ In some cases software developers are independent contractors (not employees) vis-à-vis the individual or entity for which the software is developed – *i.e.*, the software recipient. In such cases, absent a written agreement to the contrary, ownership of copyright rights in software source code created by the independent contractor for the software recipient vests, by operation of law, in the independent contractor, not the software recipient. 17 U.S.C. § 201(a) (2010). This is true even if the software recipient pays, or provides other consideration to, the independent contractor for the software. *See id.* § 202. Such payment or other consideration merely serves to pass title, from the independent contractor to the software recipient, in the material object (*e.g.*, computer diskette) in which the copyrightable software is fixed. *Id.* In order to also transfer ownership of copyright rights in the software from the independent contractor to the software recipient, those parties must execute a written agreement or other document evidencing such a transfer. *Id.* § 204(a) (“A transfer of copyright ownership, other than by operation of law, is not valid unless an instrument of conveyance, or a note or memorandum of the transfer, is in writing and signed by the owner of the rights conveyed or such owner’s duly authorized agent.”). Transfers of ownership in patentable subject matter and in federally registered marks and marks that are the subject of pending use-based registration applications are subject to similar requirements. *See* 35 U.S.C. § 261 (2010) (patents); 15 U.S.C. § 1060(a)(3) (2010) (marks).

²¹ *See* 17 U.S.C. § 202 (2010).

²² *See id.*

²³ *Id.*

C. Exclusive Copyright Rights in Computer Software.

Subject to certain statutory limitations,²⁴ the owner of copyright in an original work of authorship has the exclusive right (1) to reproduce the copyrighted work in copies; (2) to distribute such copies to the public by sale or other transfer of ownership or by lease, rent, or lending; (3) to prepare derivative works²⁵ based on the copyrighted work; (4) in the case of certain works to perform and display the copyrighted work publicly; and (5) the exclusive right to authorize others to do any of the foregoing in (1)-(4).²⁶ In the computer software context, especially as it relates to licensing of copyright rights in software, the exclusive rights described in foregoing clauses (1)-(3) and (5) most often come into play.

1. The Reproduction Right.

The “reproduction” right described in clause (1) above (“Reproduction Right”) is probably best understood by examining the following definitions of “copies,” “device,” and “machine” in Section 101 of the Copyright Act:

“Copies” are material objects, other than phonorecords, in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or indirectly with the aid of a machine or device. The term “copies” includes the material object, other than a phonorecord, in which the work is first fixed.

....

A “device”, “machine”, or “process” is one now known or later developed.²⁷

²⁴ See 17 U.S.C. §§ 107-122 (2010). Worthy of particular mention in the software context are the limitations on the “reproduction” right set forth in Section 117 of the Copyright Act, which are discussed in more detail below. See also *infra* note 28.

²⁵ A “derivative work” is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgement, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a derivative work.” 17 U.S.C. § 101 (2010) (emphasis added).

²⁶ *Id.* § 106.

²⁷ *Id.* § 101.

Thus, virtually any method in which copyrightable subject matter in computer software is duplicated, whether in photocopies, CD-ROM, binary images fixed in computer memory, or otherwise, constitutes a reproduction of such subject matter in copies and, therefore, is reserved exclusively to the owner of copyright in that subject matter and such owner's authorized designees.²⁸ Of the exclusive rights of copyright identified above, which bear most significantly on the licensing of computer software, the Reproduction Right likely is the least important of those rights for purposes of examining the differences between the closed source and open source licensing models and the impact on businesses resulting from such differences.²⁹

2. The Distribution Right.

The "distribution" right described in clause (2) above ("Distribution Right") encompasses virtually every mode and means by which copies of the copyrighted work can be passed from one individual or entity to another, as long as the work is disseminated "to

²⁸ ROSEN ON OPEN SOURCE, *supra* note 6, at 25 ("Every instance of computer software, as long as it is fixed in some tangible form, is a copy."). Certain limitations on a copyright owner's exclusive right to reproduce the copyrighted work in copies, which are set forth in Section 117 of the Copyright Act, permit those in lawful possession of an authorized copy of a computer program to further copy the program (1) "as an essential step in the utilization of the computer program" in conjunction with the computer or other device on which the program is executed; (2) to make an archival copy of the program, but only to the extent that any and all such copies are destroyed in the event that continued possession of the program is no longer authorized; and (3) in connection with certain maintenance and repair activities involving the computer or other device on which the copyrighted program is executed, but only to the extent that any and all such copies are destroyed immediately after completion of such maintenance or repair. 17 U.S.C. § 117 (2010); *see* ROSEN ON OPEN SOURCE, *supra* note 6, at 25-26.

²⁹ In the closed source licensing model, reproduction of the licensed software typically is permitted, if at all, only for copies in object code form and only as necessary for archival purposes and to permit the licensee to use the licensed software solely as permitted in the underlying license agreement. On the other hand, the open source licensing model permits licensees to copy the licensed software in source code form without restriction. *Compare* CLASSEN ON LICENSING, *supra* note 3, at 25 *with* ROSEN ON OPEN SOURCE, *supra* note 6, at 9. Thus, the more liberal approach of the open source licensing model with respect to copies of the licensed software arising out of a licensee's exercise of the Reproduction Right would not appear to directly impact that licensee's business operations unless and until the business exercised the Adaptation Right (*i.e.*, by creating derivative works of the licensed open source software) and/or the Distribution Right (*i.e.*, by distributing such software or derivative works outside the business) in connection with such copies. *See infra* text accompanying notes 30-36.

the public.”³⁰ The Distribution Right’s public dissemination aspect is often cited as the basis for re-characterizing that right as a “publication” right.³¹ As discussed in Sections I.D. and I.E. below, the nature of the control that a software licensor exerts over its licensees with respect to the Distribution Right is one of the most important distinctions between the closed source and open source licensing models.³²

³⁰ See *Hotaling v. Church of Jesus Christ of Latter-Day Saints*, 118 F.3d 199, 203 (4th Cir. 1997) (“In order to establish ‘distribution’ of a copyrighted work, a party must show that [a] . . . copy was disseminated ‘to the public.’”); 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.11[A], at 8-148 (2009) (citing *Litecubes, LLC v. N. Light Prods., Inc.*, 523 F.3d 1353, 1371 (Fed. Cir.), *cert. denied*, 129 S. Ct. 578 (2008)).

³¹ 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.11[A], at 8-148 n.1.2 (2009) (citing 17 U.S.C. § 101 (2010) (“‘Publication’ is the distribution of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease or lending.”); § 8.12[B][1], at 8-159 (2009) (“‘[D]istribution’ is the equivalent of ‘publication’”).

³² Similar to the statutory limitations on the Reproduction Right, *see supra* note 28, the Distribution Right has its own statutory limitations, the foremost of which arises out of the “first sale” doctrine. Under the first sale doctrine, the unconditional, authorized first sale or other transfer of ownership of a material object embodying an authorized copy of a copyrighted work exhausts the copyright owner’s exclusive distribution right as to that particular copy as embodied in the material object. 17 U.S.C. § 109(a); *see* 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.12[B][1], at 8-156 to 8-157 & n.13 (2009) (“Once the copyright owner consents to the sale of particular copies of his work, he may not thereafter exercise the distribution right with respect to those copies.” (citing *Omega S.A. v. Costco Wholesale Corp.*, 541 F.3d 982, 985 (9th Cir. 2008))). With respect to any copy of a copyrighted work, the first sale doctrine (1) is wholly inapplicable to a mere transfer of possession of such a copy that does not also include a transfer of ownership of such copy (*e.g.*, a transfer of possession in a licensing context), 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.12[B][1][a], at 8-157 (2009); CLASSEN ON LICENSING, *supra* note 3, at 16-17 (“Software developers, in order to avoid application of the First Sale Doctrine and retain control over redistribution of their programs, have typically distributed even mass market software under license, rather than through an outright sale, in order to prevent the First Sale Doctrine from severing control over redistribution.” (citing various cases)), and (2) does not apply to (i) any other copies of the copyrighted work that do not otherwise meet the requirements of the first sale doctrine or (ii) to any of the exclusive rights of copyright set forth in 17 U.S.C. § 106 other than the Distribution Right. *See* CLASSEN ON LICENSING, *supra* note 3, at 16-17 (citing various cases). Moreover, in recognition of the distinction between ownership of copyright rights in a work and ownership of the tangible object embodying such rights, *see* 17 U.S.C. § 202 (2010), which often comes into play in a software context, Section 109(b) of the Copyright Act, which was added via the Computer Software Rental Amendments Act of 1990, curtails application of the first sale doctrine to copyrightable subject matter in computer software by requiring that “the owner of a copy of computer software cannot lend or rent that copy to third parties without permission from the copyright owner.” CLASSEN ON LICENSING, *supra* note 3, at 17-18 (citing various

3. The Adaptation Right.

The right to prepare derivative works described in clause (3) above, which is sometimes referred to as an “adaptation” right³³ (“Adaptation Right”), reserves solely to the copyright owner of a pre-existing work (and its authorized designees) the right to prepare another copyrightable work that is based, in whole or in part, on that pre-existing work.³⁴ In order to qualify as a derivative work under the Copyright Act, a work (1) must be substantially copied from (*i.e.*, “based on”) a pre-existing work; (2) must constitute an original work of authorship in its own right (*i.e.*, must include copyrightable subject matter that differs sufficiently from copyrightable subject matter contained in the pre-existing work to satisfy the “originality” requirement for protection under the Copyright Act); and (3) must arise out some modification of the pre-existing work.³⁵ Furthermore, copyright in the derivative work exists separate and distinct from copyright in the pre-existing work on which that derivative work is based, and ownership of copyright rights in a derivative work, in and of itself, does not provide a basis for asserting ownership of copyright rights in the pre-existing work on which that derivative work is based.³⁶ Finally, copyright protection does not extend to any portion of a derivative work that is created by “unlawfully” using the pre-existing work on which such derivative work is based.³⁷ As discussed

cases).

³³ 2 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.09[A], at 8-142.8(13) (2009).

³⁴ 1 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 3.01, at 3-2, 3-3 (2009) (citing various cases).

³⁵ CLASSEN ON LICENSING, *supra* note 3, at 29-30 (citing various cases); *see* 17 U.S.C. § 101 (2010) (definition of “derivative work”).

³⁶ 17 U.S.C. § 103(b) (2010); CLASSEN ON LICENSING, *supra* note 3, at 29.

³⁷ 17 U.S.C. § 103(a) (2010); CLASSEN ON LICENSING, *supra* note 3, at 29. The district court in *Pickett v. Prince* held that plaintiff’s creation of a derivative work without consent of the defendant-owner of copyright in the preexisting work on which that derivative work is based constitutes an “unlawful” use of that pre-existing work for purposes of Section 103(a) of the Copyright Act and invalidated plaintiff’s copyright in the derivative work. 52 F. Supp. 2d 893, 904-09 & nn.17, 18 (N.D. Ill. 1999), *aff’d*, 207 F.3d 402 (2nd Cir. 2000). Section 103(a)’s unqualified denial of copyright protection for any part of a derivative work that is created via unlawful use of the pre-existing work on which that derivative work is based would appear to inject that part of the derivative work into the public domain, *see* United States Copyright Office Web Site, <http://www.copyright.gov/help/faq/faq-definitions.html> (last visited Apr. 30, 2010) (“A work of authorship is in the ‘public domain’ if it is no longer under copyright protection or if it failed to meet the requirements for copyright protection.”), even if the underlying license agreement attempts to contract around that result by vesting ownership of such derivative work in the owner of the

in Sections I.D. and I.E. below, the scope of the control that a software licensor exerts over its licensees with respect to the Adaptation Right is one of the most important distinctions between the closed source and open source licensing models.

D. The Closed Source Licensing Model.

Denying licensees access to and the right to use the source code of licensed software is the linchpin of the closed source licensing model.³⁸ Implementing this approach manifests itself in a number of different ways within the underlying license agreement, some of the more important of which are as follows:

(1) the licensor grants the licensee the right to use the licensed software, and is obligated to distribute the licensed software to the licensee, only in object code form;³⁹

pre-existing work. The author questions whether such a result is reasonable or otherwise is fair to the owner of copyright in the applicable pre-existing work. For example, if the underlying license agreement requires a licensee to assign to the owner of a pre-existing work all copyright rights in any unauthorized derivative works that are created by the licensee and based on that pre-existing work, why shouldn't such a requirement and assignment be given legal effect, notwithstanding the provisions of Section 103(a)? Or, for that matter, why shouldn't ownership of copyright in the derivative work in this situation vest by operation of law in the owner of the pre-existing work, similar to the vesting process associated with the "work made for hire" doctrine? The software licensor could attempt to "trick the system" by authorizing the licensee to make derivative works in the license agreement (even though the licensor really does not want that to occur) and require the licensee to assign all copyright rights in such works to the licensor. However, in order to facilitate such an approach, the licensor would need to provide the source code of the licensed software to the licensee, which the licensor also really does not want to do, because the right to make derivative works would carry with it an implied right to have access to such source code (even if the license agreement expressly provided otherwise). In any event, this conundrum creates a "Catch-22" situation that the software licensor should not have to confront. For an informative discussion of the foregoing issues, see 1 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 3.06 (2009).

³⁸ See generally CLASSEN ON LICENSING, *supra* note 3.

³⁹ CLASSEN ON LICENSING, *supra* note 3, at 34-35. Although the principal reason for prohibiting licensees' access to software source code in the closed source licensing model is financial (*i.e.*, closed source licensors apparently find it difficult to arrive at a source code license fee that is low enough to attract interested licensees but high enough to reasonably cover the additional risk exposure attendant to such a license), a closed source licensor may also insist on such prohibitions in an attempt to reduce the possibility of its source code falling into the hands of a competitor, *see id.* at 25, or to prevent unauthorized derivative works from passing into the public domain, *see supra* note 37.

(2) the licensee expressly is prohibited from reverse engineering, decompiling, reverse compiling, or performing any other act designed to obtain or gain access to the software source code;⁴⁰

(3) the licensee expressly is prohibited from modifying or otherwise creating derivative works of the licensed software;⁴¹

(4) the licensee is obligated to assign to the licensor all right, title, and interest, including all copyright and other intellectual property rights, in and to all derivative works of the licensed software created by or on behalf of the licensee;⁴²

(5) the licensee is obligated to secure all maintenance and support for the licensed software from the licensor or its authorized third-party vendor;⁴³

(6) in the event that the licensee is granted the right to further distribute or sublicense the licensed software, all such distributions and sublicenses cannot exceed the scope of the software license granted to the licensee – *i.e.*, the licensed software can be distributed and sublicensed in object code form only and subject to restrictions and limitations on use of the licensed software that are at least as stringent as those imposed on the licensee; and⁴⁴

(7) the licensee is obligated to prevent disclosure of and access to, and restrict the use of, the licensed software, except to the extent otherwise expressly permitted by the licensor in writing.⁴⁵

All the foregoing characteristics of the closed source licensing model (collectively, the “Closed Source Characteristics”) implicate, either directly or indirectly, the manner in which the closed source software licensor (1) relinquishes to its licensee, at least temporarily, the licensor’s exclusive dominion over the Distribution Right and Adaptation Right⁴⁶ with respect to the licensed software, but (2)

⁴⁰ *Id.* at 30-33 (citing *Bowers v. Baystate Tech., Inc.*, 320 F.3d 1317 (Fed. Cir. 2003) for the propositions that (1) software license agreement prohibitions on reverse engineering of licensed software are valid and (2) “the Copyright Act does not narrow or preempt such contractual provisions”). Some federal appellate courts, most prominently the United States Court of Appeals for the Ninth Circuit, have held that reverse engineering prohibitions in software license agreements are unenforceable in certain circumstances based on copyright “fair use” principles. For an informative discussion of some of those decisions, *see* CLASSEN ON LICENSING, *supra* note 3, at 31-33.

⁴¹ CLASSEN ON LICENSING, *supra* note 3, at 29-30.

⁴² *See id.* at 30.

⁴³ *See id.* at 34-35.

⁴⁴ *See id.* at 25-29.

⁴⁵ *See id.* at 185-92.

⁴⁶ In actuality, a closed source software licensor typically cedes nothing to its licensees in the way of Adaptation Rights, except, possibly, in the event of a release of source code to a licensee pursuant to a source code escrow arrangement. *See supra* note 4.

nevertheless, maintains close control over how the licensee exercises those rights.⁴⁷ The nature of the licensor's control in this model is quite restrictive; prohibitions abound and the licensee's obligations are directed to limiting, if not entirely abrogating, the general availability of the licensed software and any modifications thereof. For example, distribution of the licensed software is permitted only when expressly authorized in the license agreement, and, any and all such distributions must be in object code form only. Similarly, in virtually every case, the licensee is completely foreclosed from exercising the Adaptation Right.

E. The Open Source Licensing Model.

Not surprisingly, providing licensees access to and the right to use, modify, and distribute the source code of licensed software, all of which radically depart from the closed source model, are the linchpins of the open source licensing model.⁴⁸ Open source software supporters contend that such an approach more effectively furthers the objectives of software development, because (1) more individuals providing more and varied ways of analyzing and modifying software and distributing such modifications so that further modifications can occur (*i.e.*, in source code form) brings a collective brain trust to bear on software development that is absent in the closed source licensing model; (2) the responsibility for maintaining and supporting open source software is spread over many more individuals, who not only are more motivated but also have more time and resources to devote to those activities, all of which yield a more reliable software product; and (3) a more widespread and motivated community of users of a software product with access to the tools (*i.e.*, source code) required to maintain and support the product is more likely to ensure the ongoing vitality and vibrancy of that product (in order to meet the ever-increasing expectations of those who rely on the product) for a much longer duration.⁴⁹

In stark contrast to the Closed Source Characteristics, the corresponding characteristics of the open source licensing model, as reflected in the provisions of the underlying agreement under which such software is licensed,⁵⁰ are as follows (collectively, "Open Source Characteristics"):

⁴⁷ See discussion *supra* Parts I.C.2 and I.C.3 and accompanying notes.

⁴⁸ See ROSEN ON OPEN SOURCE, *supra* note 6, at 8-11.

⁴⁹ ANDREW M. ST. LAURENT, UNDERSTANDING OPEN SOURCE & FREE SOFTWARE LICENSING 6, 6-7 (2004) (hereinafter, ST. LAURENT ON OPEN SOURCE).

⁵⁰ Open source license agreements vary in type, depending primarily on the affirmative obligations imposed on the licensee for distributing any derivative works

(1) the licensor provides the licensee, and grants the licensee the right to use, the licensed software in source code form;⁵¹

(2) in view of the first Open Source Characteristic, prohibitions against obtaining the source code of the licensed software are unnecessary and otherwise not permitted;

(3) the licensee may freely create derivative works of the licensed software;

(4) in view of the third Open Source Characteristic, by operation of law the licensee owns all right, title, and interest, including all copyright and other intellectual property rights, in and to all derivative works of the licensed software created by or on behalf of the licensee;⁵²

(5) in view of the first and third Open Source Characteristics, the licensee may freely modify the source code of the licensed open source software in order to maintain (*i.e.*, correct bugs and other errors) and support (*i.e.*, update, upgrade, enhance, and further develop) that software;

(6) the licensee may freely distribute the licensed open source software and any derivative works thereof created by such licensee; however, depending upon the applicable open source code license agreement, the licensor may require any such distributions to be in source code form;

(7) in view of all the foregoing Open Source Characteristics, prohibitions against and restrictions on disclosing, accessing, and using the licensed open source software are unnecessary and otherwise not permitted; and

(8) the licensee may combine open source software and closed source software on the same storage media or in computer memory and may distribute open source software along with closed source software.⁵³

of the licensed open source software that are created by such licensee. *See* discussion *infra* Parts I.E.1, I.E.2 and accompanying notes.

⁵¹ Depending on the type of open source license agreement, the licensor may require any distributions by the licensee of the licensed source code and derivative works thereof created by the licensee to be in source code form. *See* discussion *infra* Part I.E.2 and accompanying notes.

⁵² The author submits that the open source licensing model appears to resolve the closed source licensing model conundrum, which arises out of licensees' "unlawful" creation of derivative works of licensed software resulting in such works apparently passing into the public domain, *see supra* note 37, by authorizing licensees to create such derivative works, thereby foreclosing application of Section 103(a) of the Copyright Act, and having copyright rights in such works vest in such licensees by operation of law, *see id.*

⁵³ *See* ROSEN ON OPEN SOURCE, *supra* note 6, at 8-11. "Freely" as used in the third, fifth, and sixth Open Source Characteristics means "without any conditions that would impede" the applicable activity. *Id.* at 9.

As with the closed source licensing model, all the Open Source Characteristics involve relinquishment of the licensor's exclusivity over the Distribution Right and Adaptation Right with respect to the licensed open source software. Without a doubt, the Open Source Characteristics reflect a much broader relinquishment of such exclusivity than do the Closed Source Characteristics. Nonetheless, although open source software sometimes is referred to as "free" software, such software is anything but free, primarily because open source software licensors still wield some degree of control over their licensees.⁵⁴ However, rather than emphasizing the prohibitions and restrictions typical of the closed source licensing model, such control manifests itself in the imposition on licensees of affirmative obligations designed to ensure that any exercise by such licensees of the Distribution Right and Adaptation Right maintains and, in fact, proliferates the open source nature of the licensed software.⁵⁵

License agreements under the open source licensing model often are categorized based on the type and scope of such affirmative obligations included within the agreement. Two of the most prominent of those categories for purposes of software licensed to and owned by businesses are the Academic License and the Reciprocal License, which are discussed in more detail below.

1. Academic License.

The Academic License originated with the desire of academic institutions to distribute their software to the public without the restrictions and prohibitions typical of the closed source licensing model.⁵⁶ The Academic License, which has a number of variations,⁵⁷

⁵⁴ See *id.* at 1-13. Understanding why open source software is not "free" is aided by distinguishing between the notions of "free software" and "software freedom." Compare *id.* at 13 ("Software isn't free, as in the expression 'the birds are free to fly.' Software is someone's property, and you can't use another person's property – to fly or to do anything else – without that owner's permission.") with *id.* at 3 ("Open source code is an essential requirement for software freedom, a technical prerequisite. *Software freedom* is the goal; *open source* is the means to that goal."). Software more legitimately may be thought of as "free" if it is in the public domain and thereby "[g]enerally *free* of all property claims" and available for unconditional, unlimited, and unrestricted use by anyone for any purpose. See Denise M. Everett, *Copyleft© Copyright©, Some Rights Reserved*, BENCH & B. MAG. OF THE KY. B. ASS'N, Sept. 2009, at 16-20 (emphasis added) (citing U.S. Copyright Office definition of "public domain") [hereinafter *Everett*].

⁵⁵ ROSEN ON OPEN SOURCE, *supra* note 6, at 8-11.

⁵⁶ *Id.* at 69-70.

⁵⁷ The Berkeley Software Distribution ("BSD") License, MIT (or X) License, Apache License (version 1.1), and Artistic License 2.0 are prominent examples of Academic Licenses. *Id.* at 73-102. Copies of templates of the foregoing licenses can

sets forth the licensor's notice of copyright with respect to the licensed software along with provisions by which the licensor permits licensees to unconditionally create derivative works of the software and to use the software and such derivative works in both source code form and object code form.⁵⁸ As a result, the licensor under an Academic License relinquishes exclusivity of its Adaptation Right with respect to the licensed software unconditionally – *i.e.*, without imposing on the licensee any affirmative obligations in connection with any derivative works arising out of the licensee's exercise of that right.⁵⁹ In addition, the Academic License also includes provisions that disclaim most, if not all, warranties as to, and damages arising in connection with, the licensed open source software and derivative works thereof created by the licensee.⁶⁰ Finally, most variations of the Academic License include a provision stating that the licensor does not endorse or promote any product that is derived from the licensed open source software without the licensor's prior written approval.⁶¹

In addition to the foregoing provisions, the Academic License includes a provision permitting licensees to distribute copies of the licensed software and related derivative works in either source code or object code form.⁶² However, unlike with the Adaptation Right, the licensor relinquishes exclusivity of its Distribution Right with respect to the licensed software conditionally – *i.e.*, the licensor obligates the licensee to include in such copies any and all copyright notices specified in the Academic License and to reiterate in such copies all the provisions of the Academic License.⁶³ By permitting further distribution of the licensed software and related derivative works in object code form, albeit conditionally, the Academic License opens the door for inclusion of the licensed open source software in closed source software and proprietary commercialization of such derivative works pursuant to the closed source licensing model.⁶⁴

2. Reciprocal License.

be found on the web site of the Open Source Initiative (“OSI”) at www.opensource.org, which also identifies other open source license agreements that have been approved by the OSI as conforming to the Open Source Definition set forth at <http://www.opensource.org/docs/osd>. See <http://www.opensource.org/licenses/alphabetical> (last visited Apr. 30, 2010).

⁵⁸ ST. LAURENT ON OPEN SOURCE, *supra* note 49, at 15.

⁵⁹ See *id.* at 14-18.

⁶⁰ See *id.* at 15-18.

⁶¹ See *id.* The MIT (or X) license does not include such a provision. *Id.* at 14-15.

⁶² See *id.* at 14-18.

⁶³ See ROSEN ON OPEN SOURCE, *supra* note 6, at 80-85.

⁶⁴ ST. LAURENT ON OPEN SOURCE, *supra* note 49, at 14; *Everett, supra* note 54, at 18 & n.20.

The Reciprocal License,⁶⁵ which also has a number of variations,⁶⁶ takes the rights provided to, and affirmative obligations imposed on, licensees under the Academic License a compelling step further by (i) eliminating the licensee's right to distribute copies of derivative works of the licensed software in both source code form and object code form and (ii) obligating licensees to make any such distributions in source code form only without remuneration or restriction on use.⁶⁷ As with the Academic License, the licensor's relinquishment of its Adaptation Right with respect to the licensed software under a Reciprocal License is unconditional. However, under the Reciprocal License, the conditional nature of the licensor's relinquishment of its Distribution Right with respect to the licensed software under the Academic License is reinforced by requiring that any distribution of copies of derivative works be in source code form only. This approach forecloses proprietary commercialization of any such derivative works pursuant to the closed source licensing model.⁶⁸

F. So, What's The Real Difference?

Simply stated, the different consequences (compared to the closed source licensing model) arising when an open source software

⁶⁵ The "reciprocal" moniker derives, at least in part, from the following paraphrased description of the bargain between an open source software licensor and its licensees, which is attributed to developers of one of the most influential Reciprocal Licenses, the GNU General Public License (v.2/v.3) ("GPL"): "You may have this free software on condition that any derivative works that you create from it and distribute must be licensed to all under the same license." ROSEN ON OPEN SOURCE, *supra* note 6, at 103. The Reciprocal License is also referred to as a "copyleft" license (to underscore its "liberal" departure from the copy"right" norms of the closed source licensing model) and as a "viral" license (in scornful recognition of "transmitting" the reciprocity aspect of the open source licensing model to downstream licensees). ROSEN ON OPEN SOURCE, *supra* note 6, at 105; ST. LAURENT ON OPEN SOURCE, *supra* note 49, at 38, 39; *Everett*, *supra* note 54, at 18 & n.23.

⁶⁶ Along with the GPL, the Mozilla Public License 1.1 ("MPL") is an example of another influential Reciprocal License. *See* ROSEN ON OPEN SOURCE, *supra* note 6, at 141-59; ST. LAURENT ON OPEN SOURCE, *supra* note 49, at 62-81. Copies of templates of the GPL and MPL can be found on the web site of the OSI at www.opensource.org.

⁶⁷ ST. LAURENT ON OPEN SOURCE, *supra* note 49, at 34; *see* ROSEN ON OPEN SOURCE, *supra* note 6, at 103-07.

⁶⁸ *See* ROSEN ON OPEN SOURCE, *supra* note 6, at 10 ("Under th[e] open source principle, a licensor cannot charge a royalty for the privilege to create and distribute derivative works, or require a licensee to pay a royalty for copies of a derivative work that are distributed, or impose any restriction on the type or character of those derivative works.").

licensor relinquishes exclusivity of its Distribution Right with respect to that software and related derivative works is at the very heart of any copyright-related concern that a business might have with using open source code software in connection with its business. In the worst case scenario – *i.e.*, in a Reciprocal License context – a business needs to carefully plan ahead to ensure that any derivative works of software that the business desires to own on a proprietary basis and distribute outside the business are not created based on software that is subject to such a license.

II. Is the Open Source Licensing Model Here to Stay?

A party's chronic failure to enforce its copyright rights against infringers, including a software licensee whose use of such software exceeds the scope of the underlying license, can result in loss of such rights.⁶⁹ Similarly, a party's unreasonable delay in enforcing its copyright rights against an infringer, which either (1) prejudices the infringer's ability to reasonably defend itself against any infringement claims asserted by the dilatory party (*i.e.*, "evidentiary prejudice") or (2) upsets the infringer's reasonable expectations arising out of reliance on such delay (*i.e.*, "expectations-based prejudice") can render such copyright rights unenforceable against that infringer.⁷⁰ Thus, reliable indicators of the ongoing viability of the open source licensing model may very well be (1) the regularity, dispatch, and vigor with which open source licensors attempt to enforce the affirmative obligations and other conditions imposed on their licensees, all of which are directed to sustaining and building on the open source nature of the licensed software, and (2) even more importantly, the courts' willingness to give legal effect to such enforcement efforts.

Although the open source licensing model has been on the scene for over thirty years,⁷¹ efforts to enforce open source license

⁶⁹ See *Stuff v. E.C. Publ'ns., Inc.*, 342 F.2d 143, 145 (2d Cir. 1965); 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.06, at 13-280 (80th rel. 2009).

⁷⁰ *Danjaq LLC v. Sony Corp.*, 263 F.3d 942, 950-56 (9th Cir. 2001) (describing the requisite elements of a laches defense to copyright infringement); 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 12.06[B], at 12-158 to 12-160.2 (80th rel. 2009). There is a split in circuit authority on whether laches constitutes a viable defense against copyright infringement, in view of the three-year statute of limitations period for filing civil copyright infringement actions set forth in 17 U.S.C. § 507. See 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 12.06[B], at 12-154 to 12-158 (80th rel. 2009).

⁷¹ See *Everett*, *supra* note 54, at 17 & n.16 (stating that origin of the term "copyleft" dates back to 1976).

agreements did not gain significant traction in the United States until 2007.⁷² In September of that year, the Software Freedom Law Center, an organization that provides legal services to non-profit developers of open source software (“SFLC”), filed a copyright infringement action in the United States District Court for the Southern District of New York on behalf of two developers of the software application, BusyBox (collectively, the “BusyBox Developers”), which was licensed to the defendant re-distributor, Monsoon Multimedia, Inc. (“Monsoon”), under the GPL (the “Monsoon Action”).⁷³ The complaint in the Monsoon Action alleged, among other things, that Monsoon failed to re-distribute the BusyBox product and related derivative works to end users in source code form as required by the GPL.⁷⁴

⁷² An organization known as “FSF France” was involved in open source software enforcement activities in France at least as early as 2002. *See infra* note 78. In addition, such activities in Germany spearheaded by an organization known as “gpl-violations.org” (“GPL-Violations”) and directed to alleged violations of the GPL apparently have been underway since at least as early as 2004. *See* GPL-Violations Web Site, <http://www.gpl-violations.org/about.html#whois> (last visited Apr. 30, 2010). GPL-Violations has received a number of German court rulings upholding the terms of the GPL, specifically the requirement for licensees to re-distribute licensed software and related derivative works in source code form, in the context of granting injunctions against further infringing activity. *See id.* and links thereon to case summaries under “News” (last visited Apr. 30, 2010). Moreover, in the case against D-Link Germany GmbH, GPL-Violations requested, and the court awarded, reimbursement of GPL-Violations’ legal expenses and other costs incurred in connection with suing D-Link. *See* GPL-Violations Web Site, http://www.gpl-violations.org/news/20060922-dlink-judgement_frankfurt.html (last visited Apr. 30, 2010). The action filed by GPL-Violations in a French court in late 2007 against Iliad, a French telecom company, for violating the GPL marked GPL-Violations’ first foray into a court outside of Germany (“Iliad Action”). *See* GPL-Violations Web Site, <http://www.gpl-violations.org/news/20071120-freebox.html> (last visited Apr. 30, 2010). GPL-Violations states that the Iliad Action was filed in “cooperation with . . . FSF France and the busybox authors,” *see id.*, which suggests that open source software enforcement entities in the United States and Europe may be coordinating their efforts on an international scale. As of June 2006, GPL-Violations claimed to have enforced the terms and conditions of the GPL against one hundred alleged infringers with a one-hundred percent success rate. *See* GPL-Violations Web Site, <http://www.gpl-violations.org/about.html> (“Every GPL infringement that we started to enforce was resolved in a legal success, either in-court or out of court.”) (last visited Apr. 30, 2010).

⁷³ SFLC Web Site, <http://www.softwarefreedom.org/news/2007/sep/20/busybox/> (last visited Apr. 30, 2010); *see* Wikipedia, the Free Encyclopedia (“Software Freedom Law Center”), http://en.wikipedia.org/wiki/Software_Freedom_Law_Center (last visited Apr. 30, 2010) (stating that the Monsoon Action “is believed to be the first U.S. court case in which the complaint concerned a GPL violation”).

⁷⁴ *See* Monsoon Action complaint, a copy of which is available at <http://www.softwarefreedom.org/news/2007/sep/20/busybox/complaint.pdf>, at ¶¶ 8-

In addition, later in 2007 and in 2008, SFLC filed additional copyright actions in the Southern District of New York on behalf of the BusyBox Developers, each of which (1) alleged copyright infringement of the BusyBox software product on the same grounds asserted in the Monsoon Action and (2) settled out of court.⁷⁵

Finally, in December 2009, SFLC sued fourteen consumer electronics companies in the Southern District of New York on behalf of one of the BusyBox Developers and the Software Freedom Conservancy, an organization composed of free and open source software developers including BusyBox,⁷⁶ complaining of the same GPL violation alleged in the Monsoon Action.⁷⁷

Despite this flurry of enforcement activity and resulting lawsuits in the United States in 2007 and 2008, all directed to recognizing and validating an open source licensor's right to relinquish its Distribution Right with respect to licensed open source software and related derivative works and at the same time require its

16 (last visited Apr., 30, 2010). A little over a month after the Monsoon Action was filed, the parties settled the case, and Monsoon agreed to comply with the terms of the GPL (v.2) that it allegedly violated and to pay damages to the BusyBox Developers. See SFLC Web Site, <http://www.softwarefreedom.org/news/2007/oct/30/busybox-monsoon-settlement/> (last visited Apr. 30, 2010).

⁷⁵ The defendant redistributors in those actions were Xterasys Corporation, High-Gain Antennas, LLC, Verizon Communications, Inc., Bell Microproducts, Inc., Super Micro Computer, Inc., and Extreme Networks, Inc. All those cases ultimately settled out of court. SFLC Files GPL Violation Lawsuit Against Extreme Networks, Inc. – Software Freedom Law Center (July 21, 2008), <http://www.softwarefreedom.org/news/2008/jul/21/busybox/>; BusyBox Developers Settle Case With Extreme Networks – Software Freedom Law Center (Oct. 6, 2008), <http://www.softwarefreedom.org/news/2008/oct/06/busybox-extreme-settle/>. In late 2008 SFLC also filed a copyright infringement action against Cisco Systems, Inc. on behalf of the Free Software Foundation, a non-profit organization advocating computer user freedom and committed to defending the rights of “all free software users” (“FSF”). SFLC Files Lawsuit against Cisco on Behalf of the FSF – Software Freedom Law Center (Dec. 11, 2008), <http://www.softwarefreedom.org/news/2008/dec/11/cisco-lawsuit/>; Peter Brown, *Free software is a matter of liberty, not price*, FREE SOFTWARE FOUNDATION, May 11, 2001, <http://www.fsf.org/about/>. This case settled out of court in May 2009. FSF and Cisco Settle GPL Dispute – Software Freedom Law Center (May 20, 2009), <http://www.softwarefreedom.org/news/2009/may/20/fsf-cisco-settle/>.

⁷⁶ Software Freedom Law Center, *Best Buy, Samsung, Westinghouse, and Eleven Other Brands Named in SFLC Lawsuit*, Dec. 14, 2009, <http://www.softwarefreedom.org/news/2009/dec/14/busybox-gpl-lawsuit/>. The defendants include Best Buy, Inc., Samsung Electronics America, Inc., Westinghouse Digital Electronics, LLC, and JVC Americas Corporation, among others. See Compl. at 1, *Software Freedom Conservancy, Inc. v. Best Buy Co.* (S.D.N.Y. 2009) (No. 09CIV 10155) available at <http://www.softwarefreedom.org/resources/2009/busybox-complaint-2009-12-14.pdf> [hereinafter *Best Buy Complaint*].

⁷⁷ See *Best Buy Complaint*, *supra* note 76, at ¶ 27.

licensees to provide any such distributions in source code form, a string of settlements in these lawsuits deprived United States courts of assessing and ruling on the legitimacy of the open source licensing model.⁷⁸ That all changed, however, in August 2008 with the CAFC's decision in *Jacobsen v. Katzer*.⁷⁹

In *Jacobsen*, a developer of open source software, Jacobsen, used for controlling model trains sued for copyright infringement and moved for a preliminary injunction to prevent its licensee, Katzer, from distributing copies of the licensed software without affixing to

⁷⁸ In 2002 AFPA, a French education organization, sued Edu4, a French computer equipment provider, for allegedly violating the GPL by (1) not redistributing to AFPA in source code form certain GPL-licensed open source software used by AFPA to secure remote desktop access to equipment provided by Edu4 to AFPA and (2) removing copyright and license notices from copies of the licensed software. FSF France, *Paris Court of Appeals Condemns Edu4 for Violating the GNU General Public License*, Sept. 22, 2009, <http://fsffrance.org/news/article2009-09-22.en.html>. In September 2009, the Paris Court of Appeals ruled in AFPA's favor in this case. *Id.* In an interesting contrast with the lawsuits involving the BusyBox software product, *see supra* notes 72-76 and accompanying text, the plaintiff in the French case, AFPA, is neither the owner of copyright rights in the licensed open source software nor the owner's exclusive licensee but, rather, is a non-exclusive end user sublicensee of that software. U.S. law typically permits only the owner of copyright rights, or its exclusive licensee (in the role of a licensor of such rights) or third-party beneficiary, to file suit to enforce such rights. *See* 17 U.S.C. § 501(a)-(b) (2010); *Random House, Inc. v. Rosetta Books LLC*, 150 F. Supp. 2d 613, 617 (S.D.N.Y. 2001), *aff'd per curiam*, 283 F.3d 490 (2d Cir. 2002). On the other hand, French law apparently permits non-exclusive end user sublicensees, such as AFPA, to enforce against its licensor the terms of the license agreement between its licensor and the owners of copyright in the licensed software upon any failure by the licensor to distribute the licensed software in accordance with that license agreement, so long as the licensor grants to the sublicensee rights that were granted to the licensor by those copyright owners. *See* FSF France, *Paris Court of Appeals Condemns Edu4 for Violating the GNU General Public License*, Sept. 22, 2009, <http://fsffrance.org/news/article2009-09-22.en.html>.

⁷⁹ *Jacobsen v. Katzer*, 535 F.3d 1373 (Fed. Cir. 2008). This copyright infringement action, which was brought in the United States District Court for the Northern District of California, *id.* at 1374, normally would be heard on appeal by the United States Court of Appeals for the Ninth Circuit ("Ninth Circuit"), *id.* at 1377-78. However, certain patent-related claims also were asserted by the plaintiff in the district court action, thereby giving the CAFC jurisdiction over the appeal, *id.* at 1377 (stating that, as long as a district court's subject matter jurisdiction over an action is based in any way on 28 U.S.C. § 1338(a), the CAFC has appellate jurisdiction over the action, even if no patent-related issues are appealed), subject to the CAFC looking to Ninth Circuit interpretive law on any issues not falling under the CAFC's exclusive jurisdiction, such as whether a preliminary injunction should issue, *id.* at 1377-78. Under Ninth Circuit law, when a copyright owner pursuing a preliminary injunction against copyright infringement, as opposed to a mere breach of contract claim, shows that it is likely to prevail on the merits of its infringement claim, irreparable harm is presumed. *Id.* at 1378 (citing *Perfect 10, Inc. v. Amazon.com, Inc.*, 487 F.3d 701, 713-14 (9th Cir. 2007)).

them certain notices, descriptions, and references required by the underlying Artistic License.⁸⁰ The District Court denied Jacobsen's request for a preliminary injunction, ruling that Katzer's violation of the Artistic License requirements did not constitute copyright infringement, and Jacobsen appealed.⁸¹

On appeal the CAFC commented at some length on the benefits of open source software, including that the software is "a widely used method of creative collaboration that serves to advance the arts and sciences in a manner and at a pace that few could have imagined just a few decades ago."⁸² The CAFC went on to examine the Artistic License requirements at issue and determined that "the heart of the argument on appeal concerns whether the terms of the Artistic License are conditions of, or merely covenants to, the copyright license."⁸³

After analyzing the conditional nature of the license grant in the Artistic License, relevant copyright infringement decisions from the Ninth Circuit and the United States Court of Appeals of the Second Circuit ("Second Circuit"), and contract interpretation decisions under California contract law, the CAFC concluded that the requirements violated by Katzer are conditions to that license grant, thereby giving rise to infringement of Jacobsen's copyright rights in the licensed software.⁸⁴

Although the CAFC's ruling in *Jacobsen* is in the admittedly narrow context of a preliminary injunction determination involving violations of an Artistic License, several comments by the court

⁸⁰ *Id.* at 1375-76.

⁸¹ *Id.* at 1377. The district court concluded that, at most, Katzer's failure to comply with the Artistic License requirements amounted to a breach of contract, which, in and of itself, does not provide a basis for presumed irreparable harm under Ninth Circuit law. *Id.*

⁸² *Id.* at 1378-79. The court also states that "[t]here are substantial benefits, including economic benefits, to the creation and *distribution* of copyrighted works under public licenses that range far beyond traditional license royalties." *Id.* at 1379 (emphasis added).

⁸³ *Id.* at 1379-80.

⁸⁴ *Id.* at 1380-83. In reaching its conclusion, the CAFC, among other things, (1) rejected Katzer's argument that copyright infringement cannot be premised on violation of a license grant for which consideration is exacted, not in monetary form, but, rather, in the form of conditions imposed on the exercise of the licensed rights and (2) noted that the Artistic License conditions violated by Katzer "were both clear and necessary to accomplish the objectives of the open source collaboration, including economic benefit." *Id.* at 1380-82. The CAFC vacated the district court's decision and remanded, *id.* at 1382, however, upon further consideration, the district court again denied Jacobsen's request for an injunction for failure to show irreparable harm, *see* Jason Haislmaier, *Practical Strategies for Developing Open Source Compliance Programs: Why Compliance (Increasingly) Matters*, THE PRACTICAL LAWYER, December 2009, at 47 [hereinafter *Haislmaier*].

suggest that the impact of this decision likely will be heavily relied upon and used to significant advantage by the open source community in cases whose facts and context fall well outside the confines of *Jacobsen*. For example, the CAFC states that “[c]opyright holders who engage in open source licensing have the right to control the modification and distribution of copyrighted material.”⁸⁵ Coming on the heels of the court’s decidedly favorable perspective on the open source movement,⁸⁶ such a broad-sweeping statement regarding the rights of owners of copyright in open source software not only speaks volumes about at least one appellate court’s inclinations toward enforcing such rights but also appears to legitimize perhaps the most important (and controversial) aspect of the Reciprocal License – *i.e.*, an open source software licensor’s right to relinquish exclusivity of its Distribution Right with respect to the licensed software, while at the same time controlling licensees’ distribution of the licensed software by requiring such distribution to be in source code form only.

The CAFC’s decision in *Jacobsen* is also important from a remedies perspective, because the Copyright Act provides certain remedies for copyright infringement (*e.g.*, statutory damages, injunctive relief, and other remedies) that are not (or may not be) available in connection with a mere breach of contract claim.⁸⁷ With

⁸⁵ *Jacobsen v. Katzer*, 535 F.3d 1373, 1381 (Fed. Cir. 2008).

⁸⁶ *See supra* note 84 and accompanying text.

⁸⁷ 17 U.S.C. §§ 502, 504(c) (2010). Statutory damages typically range from \$750 to \$30,000 per infringed work, as determined by the court or jury, as the case may be, to be just; however, the court or jury, as the case may be, has discretion to award statutory damages ranging from \$250 to \$150,000 per infringed work, depending on its determination as to the relative innocence (or willfulness) of the infringement. *Id.* Actual damages incurred by a copyright infringement plaintiff, which are measured by the extent to which the infringement has injured or destroyed the market value of the infringed work as of the time of the infringement, *In-Design v. K-Mart Apparel Corp.*, 13 F.3d 559, 563 (2d Cir. 1994); 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 14.02[A] (80th rel. 2009), often are difficult to prove, *see* 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 14.02[A][1] (80th rel. 2009). Therefore, because statutory damages can be recovered by a prevailing copyright infringement plaintiff without proving the actual damages incurred by the plaintiff resulting from the infringement, *see Columbia Pictures Television, Inc. v. Krypton Broad. of Birmingham, Inc.*, 259 F.3d 1186, 1194 (9th Cir. 2001); 4 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 14.04[A] (80th rel. 2009), statutory damages many times are the preferred monetary remedy for such plaintiffs that have registered the infringed work. *See supra* note 13 (registration is a prerequisite for pursuing statutory damages). In addition, in certain circumstances the Copyright Act also provides for (1) recovery of court costs; (2) award of attorney’s fees to a prevailing copyright infringement plaintiff; (3) injunctive relief; and (4) impoundment and disposition of infringing articles, 17 U.S.C. §§ 412, 505, 503, 505 (2010), none of which may be available in connection with a breach of contract claim.

this more compelling arsenal of potential remedies available under copyright law, aggrieved owners of copyright rights in open source software should be much less reluctant to enforce such rights against infringers.⁸⁸

There seems little doubt that the ever-burgeoning development and use of open source software,⁸⁹ the vigorous and seemingly well-coordinated enforcement efforts undertaken by SFLC, FSF, GPL-Violations.org, and others, and the CAFC's pivotal decision in *Jacobsen* all bode well for the ongoing vibrancy and vitality, as well as the longevity, of the open source licensing model.

III. What Should A Business Do About Open Source Software?

So far we have learned that the open source licensing model is a lot like the closed source licensing model, except in one very important respect relating to distribution of the licensed software. Unlike closed source software, which virtually never is distributed in source code form, open source software can (and sometimes must) be distributed in source code form. In addition, when licensed under an Academic License, any distribution to others of copies of licensed open source software and/or related derivative works typically must include on such copies the terms of the Academic License and certain attributions and copyright notices. Similarly, when licensed under a Reciprocal License, any distribution to others of copies of derivative works of the licensed software typically must be in source code form. Lest the temptation arise to give short shrift to the seemingly less restrictive requirements of the Academic License, one need only be reminded that violation of the requirements of an Academic License, not a Reciprocal License, formed the basis for the plaintiff's copyright infringement claim in *Jacobsen*.

We have also determined that the open source licensing model is not a "flash in the pan" approach to licensing software that is not likely to stand the test of time. On the contrary, the open source licensing model is here to stay, and organizations that either ignore or refuse to educate themselves on, and ultimately take steps to comply

⁸⁸ *Haislmaier, supra* note 84, at 47. This decreased reluctance to enforce rights in open source software translates into an increased risk for those companies that refuse to remain mindful of and respect such rights. *See id.*

⁸⁹ ROSEN ON OPEN SOURCE, *supra* note 6, at 313 ("The open source paradigm is transforming software development and distribution around the world. More and more consumers, companies, and government agencies are now demanding that they be allowed software freedom" in accordance with the principles of the open source licensing model.).

with, the requirements of that model, likely do so at their peril. Thus, organizations that embrace open source software from an educated perspective – *i.e.*, by identifying and assessing the potential benefits of such software in view of the potential risks to the organization arising from use of the software – are likely to make the most prudent and uneventful transitions to the open source licensing model.

Informed by these determinations, we can now address the questions asked at the beginning of this article. It is important at the outset to highlight two significant factors likely to shape an organization's overall approach to open source software: (1) the organization's need to own (rather than license from others) and distribute its software-related assets and (2) the organization's stage of development.

Clearly, any organization that needs to own and distribute its software-related assets in order to achieve its business objectives would not be a likely candidate for transitioning to the open source licensing model. For example, a software development organization, whose internally developed software is a core asset that is licensed or otherwise distributed to customers, and from which derivative works are developed, on a regular basis, likely will require a very thorough, detailed, and comprehensive plan for any transition to the open source licensing model, so as to preserve the ongoing value of that asset. For that matter, such an organization's use of open source software could be entirely out of the question. The same would hold true for any entity whose value proposition is based in any way on obtaining and maintaining ownership of copyright rights in software-related assets.

At the other end of the spectrum, an organization that owns no software, does not plan to ever own any software and is perfectly happy to license all its software from other entities, and has no need to distribute the software it uses outside the organization should be able to transition to the open source licensing model with relative ease.

Similarly, the often compelling financial, "time to market," and other unique challenges facing earlier stage organizations likely would make the low cost, flexibility, and enhanced availability of open source software hard for such a company to resist, notwithstanding any adverse consequences to the organization that might arise out of using such software.⁹⁰ In the event that the organization has no overriding business need to own and externally distribute, and is content licensing from others, its software-related assets, it should be

⁹⁰ This is not to say that open source software would not have appeal to mid-stage and late stage companies as well. For example, such companies may find open source software useful in maintaining a competitive advantage in really tight markets or introducing a level of efficiency and effectiveness in the organization's business operations that cannot be achieved any other way.

able to hop on the open source bandwagon without ever looking back. However, for a startup software developer, for example, whose proprietary software is its lifeblood, giving into the temptation to cut cost and time to market by incorporating open source software into its product could be the worst business decision that it ever makes.

A. Does (or Should) the Organization Use Open Source Software in its Business?

An organization should consider taking the following initial steps in formulating an approach for addressing open source software issues: (1) review and revamp the organization's strategy on how software-related assets are used, and will be used for the foreseeable future, within the organization to take account of the unique issues associated with open source software and (2) audit the organization's existing software-related assets to (i) determine whether and to what extent open source software is being used in connection with the organization's business activities; (ii) identify the purposes for which such software is used; (iii) assess whether such uses are compatible with the organization's revamped software utilization strategy; and (iv) make adjustments as needed to bring about the desired level of compatibility.

1. Review and Revamp Software Utilization Strategy.

Knocking the dust off the organization's software utilization strategy and breathing new life into it by looking for sensible and beneficial ways to incorporate open source software into achievement of the organization's business objectives is essential to successfully transitioning to the open source model. All business units within the organization should take part in this process in order to draw from as broad an experiential base as possible and to ensure that all necessary expertise is brought to bear on this important undertaking. In some cases, it may be necessary to bring in outside consultants having specialized expertise that may be lacking within the organization.

Tapping into expertise in at least the following areas is a must: (1) *technical* – *i.e.*, individuals having in-depth understanding of (i) the operational aspects and functional capabilities of the organization's software-related assets (typically, the organization's information technology ("IT") department) and (ii) whether and to what extent the organization's use of open source software will enhance its business operations; (2) *risk management* – *i.e.*, individuals who can determine the scope of additional potential liability to which the organization will be exposed by transitioning to open source software; (3)

product/service development – i.e., individuals who can assess the potential impact of an open source transition on the demand for the organization's product and service offerings; (4) *financial* – i.e., individuals who can evaluate how an open source transition likely will affect the organization's bottom line; (5) *human resources* – i.e., individuals who can determine whether and to what extent an open source transition (i) will be embraced by the organization's employees and (ii) will impact the organization's culture; and (6) *contract administration* – i.e., individuals who can identify and develop methods for addressing contract administration issues that may arise out of an open source transition. In the event that the organization's contract administrators or other personnel do not possess appropriate expertise in copyright law and the intricacies of intellectual property license agreements, it may be necessary to engage intellectual property counsel to assist in those areas.

Some of the more important points to consider in connection with revamping the organization's software utilization strategy to accommodate open source software are as follows:

(1) Which software-related assets are owned (or licensed) and likely will remain owned (or licensed) by the organization for the foreseeable future?

(2) What factors does the organization take into account in deciding whether to own or license a software-related asset?

(3) How frequently, if at all, does the organization prepare or have prepared derivative works of any of the software-related assets that it owns?

(4) How important to achieving the organization's business objectives is developing and maintaining ownership of such derivative works?

(5) How frequently, if at all, does the organization distribute to third parties (i) any software-related assets that it owns or (ii) derivative works of any such assets?

(6) How important to achieving the organization's business objectives is distributing and maintaining ownership of such assets and derivative works?

(7) Is the software distributed by the organization self-contained, imbedded in other software, or both?

(8) What are the perceived benefits, if any, of transitioning to open source software?

(9) What are the perceived risks, if any, of transitioning to open source software?

(10) Which aspects of the organization's business operations will be impacted by transitioning to open source software and how significant is the impact in each case?

(11) Is the organization positioning itself, or is there a reasonable possibility that the organization will position itself in the foreseeable future, to be acquired? If so, what aspects of the organization's revamped software utilization strategy need special attention in order to enhance the organization's appeal as a target?

(12) Is the organization considering changing its business model or any other significant characteristic of its business, and, if so, what aspects of the organization's revamped software utilization strategy need special attention in order to facilitate that change?

(13) Does the organization's existing software utilization strategy lend itself to being revamped to accommodate open source software or should an entirely new plan be developed?

(14) What steps should the organization take presently in order to minimize any adverse consequences relating to open source software in the future?

The questions posed in clauses (5) and (6) above implicate perhaps the most important activity within an organization for purposes of determining the advisability of transitioning to open source software – *i.e.*, “distribution” of software-related assets. Prior to OSI's approval of the GNU Affero General Public License v.3 (“AGPL”)⁹¹ as an open source license in March 2008,⁹² the term “distribute,” as used with respect to a party providing open source code software to another party, generally was understood to mean nothing more than delivery (*i.e.*, via transfer of possession) of a copy of the source code of the software fixed in any medium (*e.g.*, on disc).⁹³ Thus, a licensor of open source software to which users were provided access pursuant to an Application Service Provider (“ASP”) model over a computer network, sometimes called “Software as a Service” (“SaaS”),⁹⁴ typically was not obligated to provide such users

⁹¹ GNU Affero General Public License v3 – Open Source Initiative, <http://www.opensource.org/licenses/agpl-v3.html> (last visited Apr. 30, 2010) (providing a copy of the AGPL).

⁹² *Press Release, Funambol, Funambol Helps New GPLv3 Open Source License Gain Formal OSI Approval (March 13, 2008)*, http://www.funambol.com/news/pressrelease_2008.3.13.php.

⁹³ *Affero General Public License – Wikipedia, the free encyclopedia*, http://en.wikipedia.org/wiki/Affero_General_Public_License (last visited Apr. 30, 2010). The provision requiring that SaaS governed by the AGPL be provided in source code form is referred to as the “Affero clause,” “Affero requirement,” and “Affero provision.” *Id.* The FSF recommends that developers use the AGPL for any software provided over a network. Various Licenses and Comments about Them – GNU Project – Free Software Foundation (FSF), <http://www.gnu.org/licenses/license-list.html> (last visited Apr. 30, 2010).

⁹⁴ *See Application service provider – Wikipedia, the free encyclopedia*, http://en.wikipedia.org/wiki/Application_Service_Provider (last visited Apr. 27, 2010).

the licensed open source software and related derivative works in source code form, because such software was not “distributed” in the manner required to trigger such an obligation.⁹⁵

However, with the advent of the AGPL, which “close[d] a perceived application service provider ‘loophole’ . . . in the ordinary GPL,” a licensor of open source SaaS governed by the AGPL must provide its end users with copies of the SaaS and related derivative works in source code form.⁹⁶ Consequently, in examining the questions posed in clause (5) and (6) above, an organization that provides SaaS in connection with its business should remain particularly mindful of the requirements for using open source software governed by the AGPL.

2. Open Source Software Audit.

Regardless of whether the organization revamps its software utilization strategy to accommodate open source software, the organization should audit its software-related assets to determine whether and to what extent open source software is used in its business operations. This is important for at least two reasons:

First, even if the organization’s revamped software utilization strategy abstains from, or significantly limits, the use of open source software, there are no guarantees that open source software has not already found its way onto the organization’s computer system. Thus, identifying any such open source software and promptly locating and installing closed source replacement software as needed to reconcile with the revamped strategy will provide the organization peace of mind going forward.

Second, if the revamped strategy embraces the use of open source software in a significant way, it will be necessary to determine whether and to what extent any open source software presently used by the organization is compatible with that strategy.

There are a number of vendors that provide services for auditing open source software. In cases where cost is a concern or an outsider’s access to the organization’s software and underlying code presents too much potential risk, even with a confidentiality agreement in place, an organization may elect to use its own IT department to conduct the audit. In any event, the organization should undertake appropriate due diligence and planning with respect to the entities and individuals that will conduct the audit to ensure that (1) they possess

⁹⁵ See Affero General Public License – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Affero_General_Public_License (last visited Apr. 27, 2010).

⁹⁶ *Id.*

the requisite expertise; (2) their products are capable of auditing the organization's software in a manner designed to achieve the organization's objectives for the audit; and (3) they are subject to appropriate written confidentiality and restricted use obligations. Following are some of the more important objectives that an organization may want to consider in this context:

(1) Audit scope should include all software (*i.e.*, system, application, and programming software) on the organization's computer system.

(2) Audit results should identify the precise origin (*i.e.*, licensor(s)) of each item of open source software and the license agreement governing use of that item.

(3) Audit results should identify all software governed by an open source license agreement, even if that software resides on the organization's computer system in object code form only.⁹⁷

(4) Audit results should identify all open source software that is electronically distributed⁹⁸ outside the organization.

(5) Audit results should separately identify open source software that is linked to or incorporated within any of the organization's closed source software.

(6) In the event that the audit reveals any open source software that is not compatible with the organization's revamped software utilization strategy, the vendor should be able to recommend functionally-equivalent closed source software with which to replace the incompatible open source software.

B. Planning for the Future with Open Source Software.

Once the organization revamps its software utilization strategy to take account of open source software, audits its software to establish a baseline for current usage of open source software within the organization, and reconciles the results of the audit with the objectives of the strategy, it's time to plan for the future. The goal here is to allow the organization to reap as many benefits as possible from using open source software without exposing itself to an undesirable scope of potential liability in the process. Following are some steps that the organization should consider in connection with its planning activities:

⁹⁷ Software governed by an Academic License may be distributed in either source code or object code form. *See supra* note 58 and accompanying text.

⁹⁸ For software governed by the AGPL, it is important to remember that "distribution" encompasses provision of network access to licensed software via SaaS as well as other, more conventional distribution methods. *See supra* notes 91-95 and accompanying text.

(1) All proposed uses of open source software throughout the organization should funnel through a person or persons, who are tasked with assessing whether such use is compatible with the organization's software utilization strategy. This assessment team should comprise individuals, or at a minimum solicit input, from the same constituencies involved in reviewing and revamping the organization's software utilization strategy. The assessment, which should be conducted according to a protocol made available to all the organization's employees, would serve as a filter but not be so rigid as to reject proposed uses of open source software that may be of significant benefit to the organization at the expense of incurring a bearable amount of additional risk. It is important that some degree of flexibility be permitted, not only with the organization's software utilization strategy but also with this assessment process.

(2) The organization should consider auditing all its software-related assets periodically (perhaps, once a year) along the lines described above in order to determine the level of compliance with the organization's software utilization strategy with respect to open source software. Uses of open source software that have not been approved by the assessment team or otherwise deviate from the organization's software utilization strategy promptly should be investigated to identify and address the circumstances surrounding such use – *e.g.*, who are the users, when did such use commence, where did the open source software originate, was the open source software modified or distributed outside the organization, why wasn't the use approved in advance? In addition, remedial measures should be undertaken to reconcile the results of each audit with the objectives of the software utilization strategy.

(3) Contract administration or other appropriate personnel within the organization should be required to maintain a file containing current copies of all license and other agreements governing open source software used within the organization. Each such agreement should be closely examined to determine the organization's duties and obligations thereunder, especially as relates to modifications to, and distribution outside the organization of, the licensed software and related derivative works. In the event of any uncertainty or confusion about the provisions of any such agreement, the organization should consider consulting an intellectual property attorney familiar with open source software license agreements. Two items are of particular concern here:

(i) The applicable license agreement may include a provision allowing the licensor to modify the terms and conditions of the agreement without notice to the licensee, with such modifications taking effect upon the licensee's first use of

the software after the modifications are made. For any agreement having such a provision, the organization should consider ceasing use of the corresponding software or replacing it with other software that is not subject to such a provision.

(ii) Much open source software, and the license agreements governing its use, are made available, accessed, and downloaded via the internet. While quite convenient, this creates a trap for the unwary by making it too easy to start using the software without (a) downloading a copy of the applicable license agreement or (b) securing the assessment team's pre-approval for such use. Therefore, the assessment protocol described above should highlight this point and emphasize the need for deliberation and restraint when encountering open source software on the internet.

(4) Each and every software-related agreement that the organization negotiates – whether for incoming or outgoing software – should take into account that the software at issue may include open source software. For example, agreements for bringing software into an organization⁹⁹ whose software utilization strategy precludes the use of open source software should require the software provider to represent and warrant that such software does not and will not contain open source software. Similarly, agreements for bringing software into an organization whose software utilization strategy limits use of open source software in some way (*e.g.*, only software subject to an Academic License) should include representations and warranties by the software provider to that effect. In cases where the software recipient has to be absolutely sure that the incoming software either contains no, or only certain types of, open source software, the applicable agreement should require an audit of the software along the lines described above and certification by the software provider that it meets the software recipient's expectations in this regard. For outgoing software, the roles will be reversed, and the organization, in the position of a software distributor, likely will be expected to make such representations and warranties and provide for such an audit. Moreover, to the extent of any open source included in the outgoing software, the organization's distribution of that software could trigger affirmative obligations of the type described in Part I.E. above. Thus, it is in the organization's best interest to know exactly what, if any,

⁹⁹ For example, software license agreements, software development agreements, and agreements, such as an asset purchase agreement, pursuant to which an organization acquires software-related assets.

open source software is contained in the outgoing software, which can be accomplished by subjecting the software to an open source audit.

(5) The organization's approach to planning for the future with open source software should include a forward-looking dimension. Each decision that the organization makes about open source should, among many other things, ask and answer the following questions: How will this decision impact the organization's stature as an acquisition target – *i.e.*, the organization's "marketability" so to speak? How will this decision affect the marketability of the organization's goods and services? For example, charting a vast open source software course for an organization whose most likely acquirers eschew the open source licensing model could have disastrous consequences in the event that the organization is positioning, or reasonably could position, itself to be acquired. Similarly, it may not be prudent to make a significant transition to open source software in connection with an organization's software-related product and service offerings when an attractive potential market for those products and services is occupied by end users that would resist further distributing those items and related derivative works in source code form.

IV. Conclusion

Two adages quickly should come to mind as an organization contemplates life with open source software:

- (1) Knowledge is power.
- (2) What you do not know may hurt you.

Any organization that utilizes software in connection with its business should devote the time and resources to educating itself about the potential benefits and risks of using open source software. At the end of the day, open source software might not be a good fit. On the other hand, an organization's well-thought-out, methodical approach to investigating the advisability of using open source software may uncover a valuable asset, which, with proper oversight, could enhance the organization's performance for years to come.

