DE-BUGGING OPEN SOURCE SOFTWARE LICENSING

Robert W. Gomulkiewicz[*]

Given enough eyeballs, all bugs are shallow.[1]
Eric S. Raymond

I. INTRODUCTION

Home computer users and businesses often rely on software developed by unconventional programmers[2] known as "hackers."[3] Hackers claim that the code they develop is superior in quality to the code developed by commercial software firms because hackers freely share the code they develop.[4] This code sharing enables a multitude of programmers from around the world to rapidly find and fix bugs.[5] The legal mechanism that enables hackers to deploy this

* Mr. Gomulkiewicz is Director of the Intellectual Property Law and Policy Graduate Program and Research Associate Professor of Law at the University of Washington School of Law. Formerly, he was an Associate General Counsel with Microsoft Corporation and chair of the UCITA Working Group of the Business Software Alliance. The views expressed in this article are the personal views of the author. The author would like to thank Tom Quinn for his insights into the open source community, and Carolyn Quinn and Andrea Lairson for listening patiently to many hours of discussion on the subject.

1. Eric Steven Raymond, *The Cathedral and the Bazaar*, *at* http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html [hereinafter Raymond, *The Cathedral and the Bazaar*] (last visited Sept. 22, 2002).

2. Open source leader Bruce Perens refers to hackers as "unconventional programmers." *See* Bruce Perens, *The Open Source Definition*, *in* OPENSOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION 171, 173 (Chris DiBona et al. eds., 1999), *available at* http://www.netlibrary.com/ebook_info.asp?product_id=24215 [hereinafter Perens, *The Open Source Definition*] (last visited Sept. 22, 2002).

3. Developers who have a passion for exploring the details of programming call themselves "hackers." Hackers distinguish themselves from "crackers"—programmers who use their craft for mischief and malicious purposes. ERIC S. RAYMOND, THE NEW HACKER'S DICTIONARY 233-34 (3d ed. 1996).

4. Eric S. Raymond, *The Magic Cauldron*, *at* http://www.tuxedo.org/~esr/writings/magic-cauldron/magic-cauldron.html [hereinafter Raymond, *The Magic Cauldron*] (last visited Sept. 22, 2002) (discussing advantages of open source programs); *but see* Stuart Zippers, *Linus Caves Into Criticism*, *Releases 2.4 Kernel*, LINUXGRAM, Jan. 18, 2001 (reporting that a bug-laden Linux 2.4.0 Kernel would be released).

5. *See* Raymond, *The Cathedral and the Bazaar*, *supra* note 1; Robert Lemos, *Software Flaw Threatens Linux Servers*, CNET NEWS.COM, Nov. 28, 2001, *at* http://news.com.com/2100-1001-276328.html (last visited Dec. 20, 2002).

worldwide team of de-buggers is a license agreement[6] or, to be more precise, an assortment of license agreements[7] known as "open source" licenses.[8]

Although open source software developers may regularly fix[9] buggy software, they do not regularly fix their licenses. There are a multitude of licenses that purport to meet the goals of open source development.[10] These licenses reflect different, and sometimes contradictory, approaches to core licensing issues. Many of these licenses are buggy—out of date,[11] misapplied,[12] misunderstood[13] and hopelessly confusing.[14] This state of affairs

---

6. "To stay free, software must be copyrighted and licensed." Debian GNU/Linux, *What Does Free Mean? Or What Do You Mean by Free Software?*, *at* http://www.debian.org/intro/free [hereinafter Debian GNU/Linux, *What Does Free Mean?*] (last visited Dec. 20, 2002); *see also* FREE SOFTWARE FOUNDATION, *The GNU General Public License*, pmbl., *at* http://www.fsf.org/licenses/gpl.txt [hereinafter FSF, *GNU-GPL*] (last visited Dec. 20, 2002).

7. *See generally* Robert W. Gomulkiewicz, *How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2B*, 36 HOUS. L. REV. 179 (1999) [hereinafter Gomulkiewicz, *How Copyleft Uses License Rights*] (discussing the nature of open source licenses).

8. *See* Daniel B. Ravicher, *Facilitating Collaborative Software Development: The Enforceability of Mass-Market Public Software Licenses*, 5 VA. J.L. & TECH. 11, ¶ 72 (2000), *at* http://www.vjolt.net/Vol5/issue3/v5i3a11-Ravicher.html (last visited Dec. 20, 2002); David A. McGowan, *Legal Implications of Open-Source Software*, 2001 U. ILL. L. REV. 241, 254 (2001). These licenses are also called "free software," "copyleft," "public" or "community" licenses. I use the term "open source" to refer to all these licenses. *See* OPEN SOURCE INITIATIVE, HISTORY OF THE OSI, *at* http://www.opensource.org/docs/history.php [hereinafter OSI, *History of the OSI*] (last visited Dec. 20, 2002) (stating that the "open source" terminology has emerged as the majority choice among hackers).

9. Eric S. Raymond, *How To Become a Hacker*, *at* http://www.tuxedo.org/~esr/faqs/hacker-howto.html [hereinafter Raymond, *How to Become a Hacker*] (last revised Aug. 15, 2002) ("In this imperfect world, we will inevitably spend most of our software development time in the debugging phase.").

10. *See* discussion *infra* Part III.

11. *E.g.*, the latest version of the Free Software Foundation's General Public License, version 2.0, was published in 1991. FSF, G*NU-GPL*, *supra* note 6. A new version has been discussed in hacker circles but has yet to appear. *See* Nikolai Bezroukov, *Social Aspects of the BSD vs. GLP Debate: The Catalog of Software Licenses*, http://www.softpanorama.org/Copyright/catalog_of_software_licenses.shtml [hereinafter Bezroukov, *The Catalog of Software Licenses*] (last visited Dec. 20, 2002); Mike Downing, *Revision 3.0 of open-source GPL stirs concern in the embedded space*, INTEGRATED COMMUNICATIONS DESIGN, Feb. 2002, *available at* http://www.icd.pennnet.com/ (last visited Nov. 29, 2002) (reporting Version 3.0 of the GPL is on its way).

12. Nikolai Bezroukov, *BSD vs. GPL: A Framework for the Social Analysis*; § 1.5 (Draft version 0.90), *at* http://www.softpanorama.org/Copyright/License_classification/index.shtml [hereinafter Bezroukov, *BSD vs. GPL*] (last visited Dec. 20, 2002) (noting that a disadvantage for programmers in selecting the GPL is the lack of uncertainty about whether "all side effects of GPL v.2 are already known").

13. *See* Bezroukov, *The Catalog of Software Licenses*, *supra* note 11 (suggesting that the GPL's fuzziness is not accidental).

14. *Id.* (listing and discussing advantages and disadvantages of over 30 different types of open source licenses). To quote one hacker:

> And I thought one of the reasons to use Free Software was to avoid complex licensing issues? The GPL seems to have as much controversy as any other license, whether or not it is even enforceable

benefits no one. Hackers suffer because they do not know which license form to use.[15] End users suffer because they do not fully understand the terms of use.[16] Commercial software developers suffer because they have difficulty discerning how open source licensed software may affect their intellectual property.[17]

The key to successfully de-bugging open source licensing is setting up a better process for creating and updating open source licenses. This article outlines one such process. This article begins by describing the array of open source licenses. It then explains the significant shortcomings in these licenses. The article concludes by proposing that a standards organization assume responsibility for improving important open source license forms and licensing practices.

II. Open Source Software

Software developers have shared code with each other throughout the history of software development.[18] Sharing code allows programmers to collaborate on projects and share programming ideas.[19] Programmers share code in both its human-readable source code and machine-readable object

---

> hasn't been proven yet in court. I hear all the time that using Linux "avoids all the licensing crap of MS." Well, after reading this, I'm not so sure about that. The GPL seems just as complex and irritating, if not more, than any of those lame EULA I click "agreeing" to.

Discussion posting of "Anonymous Coward," *Attorney Dan Ravicher on Open Source Legal Issues* (June 5, 2001 at 9:21 A.M.), *at* http://interviews.slashdot.org/comments.pl?sid=12603&threshold=1&commentsort =0&tid=123&mode=thread&pid=0 [hereinafter Ravicher, *Slashdot Discussion*] (last visited Dec. 20, 2002). *See also* Slashdot Discussion, Discussion Posting of Tony, *id.* (June 5, 2001 at 10:58 A.M), *at* http://interviews.slashdot.org/comments.pl?sid=12603&threshold=1&commentsort=0&tid= 123&mode=thread&pid=0 (last visited Dec. 20, 2002) ("Of course the GPL is complex. IP law is complex, the GPL was designed for one purpose—to short-circuit IP law. To do that, it has to address every possible point of IP law."). The Slashdot website is a popular hacker forum. Its slogan is: "News For Nerds. Stuff That Matters." Robert Lemos, *Behind the Slashdot Phenomenon*, CNET NEWS.COM, June 24, 2002, *at* http://news.com.com/2102-1082-938615.html (last visited Dec. 20, 2002).

15. *See infra* Part III.B.3; *cf.* Bezroukov, *BSD vs. GPL*, *supra* note 12, § 1.5 ("The widely held misconception is that the GPL is a simple license. This is simply not true.").

16. *See infra* Part III.B.

17. *See* Stephen Shankland, *BSDi Unix buy reshapes open source*, CNET NEWS.COM, Apr. 4, 2001, *at* http://news.com.com/2100-1001-255328.html (last visited Dec. 20, 2002) (issues expressed by Wind River); MICROSOFT LICENSING, MICROSOFT CORP., *The Microsoft Shared Source Philosophy*, Feb. 8, 2002, *at* http://www.microsoft.com/resources/sharedsource/default.mspx (last visited Dec. 20, 2002) (presenting positions of Microsoft on shared source software).

18. *See* Gomulkiewicz, *How Copyleft Uses License Rights*, *supra* note 7, at 182-85 (discussing the history of the open source movement).

19. *Id.*

code forms.[20] Source code is the collection of instructions a programmer writes in a given programming language to tell a computer what to do. Using a tool called a compiler, a programmer converts source code into instructions a computer can execute.[21] These executable instructions are called binary or object code.

Although it is common for developers, including commercial developers, to share source code,[22] it is also common practice to hold source code[23] as a trade secret.[24] For example, mass market software developers often treat the source code of their core products as the crown jewels of the company.[25] They do this because their customers seldom need or want[26] source code and competitors might gain free rider advantages from access to it.[27]

The hacker community, however, believes passionately that source code should be shared. The underlying rationale for this position differs among hackers. Some hackers believe that sharing code is a desirable best practice that allows programmers to create the highest quality software.[28] Others

20. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983). Source and object code are both protectable by copyright, as well as by trade secret law. *See id.* at 1253-54. *See generally* Robert W. Gomulkiewicz, *Legal Protection for Software: Still a Work in Progress*, 8 TEX. WESLEYAN L. REV. 445 (2002) (describing how software is protectable by copyright, patent, trademark, trade secret, and contract law).

21. *See* BILL GATES, THE ROAD AHEAD 24-29 (1995) (describing binary code).

22. *See* MICROSOFT LICENSING, MICROSOFT CORP., *supra* note 17 (describing Microsoft's shared (open) source initiative).

23. Some businesses treat object code as a trade secret as well, e.g., for vertical software that gives them a comparative advantage.

24. Maureen A. O'Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 DUKE L.J. 479, 493-94 & n.56 (1995).

25. *See* Robert W. Gomulkiewicz & Mary L. Williamson, *A Brief Defense of Mass Market Software License Agreements*, 22 RUTGERS COMPUTER & TECH. L.J. 335, 359 (1996). While keeping core product source code secret, many commercial publishers publish significant amounts of other source code without confidentiality restrictions. *See* MICROSOFT LICENSING, MICROSOFT CORP., *supra* note 17.

26. Some customers do need source code and Microsoft, for one, has expanded its source licensing to accommodate this. *See* MICROSOFT LICENSING, MICROSOFT CORP., *supra* note 17; Florence Olsen, *Microsoft Gives Researchers and Students Access to Source Code for Web Platform*, CHRON. HIGHER EDUC., Apr. 12, 2002, at 38.

27. *See* Ravicher, *supra* note 8, ¶ 4 (discussing the free rider problem).

28. OSI, *History of the OSI*, *supra* note 8.

maintain that code sharing increases competition.[29] Still others believe that free code is a right akin to free speech.[30]

III. OPEN SOURCE LICENSES

A. Why Hackers Use Licenses

The "open source" nomenclature leads many people to believe that hackers place their software in the public domain or sell it free of charge[31] as a copyright "first sale."[32] However, most hackers use a different transaction model—licensing. Through licensing hackers maintain control over the code they develop.[33] To put it another way, licensing allows hackers to provide others with exactly the right amount of freedom to use their code—too much freedom or too little freedom thwart the aims of open source advocates.[34]

---

29. *See* Bruce Perens, *Free Software Leaders Stand Together*, *at* http://perens.com/Articles/Stand Together.html (last visited Dec. 20, 2002). With some exceptions, open source software is becoming a commercial success. *See* Paul Festa, *Nations uniting for open source*, ZDNet, Aug. 29, 2001, *at* http://zdnet.com.com/2100-1106-530598.html (last visited Dec. 20, 2002); Matthew Broersma, *Queen dismisses Linux*, CNET NEWS.COM, Dec. 5, 2001, *at* http://news.com.com/2100-1001-276666.html (last visited Dec. 20, 2002); Sergio G. Non, *Got Linux? Many Companies Say No*, CNET NEWS. COM, Nov. 7, 2001, *at* http://news.com.com/2100-1001-275504.html (last visited Dec. 20, 2002); Matthew Broersma, *Torvalds says Linux is still suffering growing pains*, CNET NEWS.COM, Nov. 27, 2001, *at* http://news.com. com/2100-1001-276230.html (last visited Dec. 20, 2002).

30. *See* Richard Stallman, *The GNU GPL and the American Way*, *at* http://www.gnu.org/ philosophy/gpl-american-way.html (last visited Dec. 20, 2002).

31. Hackers say that free software is a matter of liberty not price. The Free Software Foundation likes to say: think "free speech," not "free beer." *See* FREE SOFTWARE FOUNDATION, *The Free Software Definition*, *at* http://www.fsf.org/philosophy/free-sw.html [hereinafter FSF, *The Free Software Definition*] (last visited Dec. 20, 2002); *see also* Tim O'Reilly, *The Open-Source Revolution*, RELEASE 1.0, Nov. 1998 [hereinafter O'Reilly, *The Open-Source Revolution*] (quoting Richard Stallman) (on file with author).

32. A "first sale" is not a sale of the copyright itself, but rather the "first sale" gives the buyer of a copy certain limited rights as to that particular copy, primarily the right to distribute the copy to someone else. Copyright Act, 17 U.S.C. § 109(a) (2000); Parfums Givenchy, Inc. v. Drug Imporium, Inc., 38 F.3d 477, 480 (9th Cir. 1994) (finding that the "first-sale" doctrine permits the owner of a lawfully purchased copy "to sell or distribute" that particular copy without interference). A first sale is really more akin to a limited license.

33. Gomulkiewicz, *How Copyleft Uses License Rights*, *supra* note 7, at 186 (explaining why hackers use licenses). *See generally* Gomulkiewicz & Williamson, *supra* note 25, at 352-56 (explaining why most software publishers use licensing rather than copyright first sales as a transaction model in the mass market); Robert W. Gomulkiewicz, *The License Is the Product: Comments on the Promise of Article 2B for Software and Information Licensing*, 13 BERKELEY TECH. L.J. 891, 895-904 (1998) [hereinafter Gomulkiewicz, *The License Is the Product*] (discussing the value of mass market licensing); Raymond T. Nimmer, *Licensing in the Contemporary Information Economy*, 8 WASH. U. J.L. & POLICY 99 (2002).

34. Although hackers use licenses, some hackers view the license as more of a social contract than a legal document. *See* Bezroukov, *BSD vs. GPL*, *supra* note 12, § 1.3 (licenses reestablishing the 'rules of

Some hackers reason that if they place their code in the public domain, someone else could effectively make their code "closed source" by preparing a derivative work and refusing to license the source code of the derivative work.[35] A "first sale" does not achieve the goals of open source development for a different reason. A copyright first sale grants the purchaser of a copy only a limited right to use and distribute that copy; it does not grant the purchaser broad rights to make and distribute derivative works of the copy.[36] This excessively limited grant of rights does not allow programmers to freely change and distribute derivative code, which are rights fundamental to open source development.[37]

B. The Buggy State of Open Source Licensing

Many licenses are used with open source software. Each of these licenses takes a different approach to core licensing issues.[38] Many of these agreements have not been updated in years.[39] Many of them are misunderstood[40] and misapplied.[41] Many of them do not work well with one

---

the game'). Even so, the debate over whether the GPL or BSD-style License is better is often very heated. *See* Joe Barr, *Live and let license*, LINUXWORLD.COM, May 23, 2002, *at* http://www.itworld.com/AppDev/350/LWD010523vcontrol4/ (last visited Dec. 20, 2002) ("The holiest of holy wars are not fought over word processors, operating systems, or compilers. They are all about software licenses."); *see also* Perens, *The Open Source Definition*, *supra* note 2, at 175-76 (describing the furor over the use of a non-open source license in conjunction with otherwise open source KDE software).

35. However, the BSD License does permit a developer to make derivatives which are "closed source." *See infra* Part III.B.4.

36. *See* Copyright Act §§ 106, 117.

37. *See infra* Part III.B.1.

38. *See infra* Part III.B.

39. *See* discussion and sources cited *supra* note 11.

40. Traditional contract interpretation analysis has an answer for ambiguous contract language: construe the ambiguity against the drafter. RESTATEMENT (SECOND) OF CONTRACTS § 206 (1981). However, the intellectual property overlay on license contracts alters this rule of contract construction. *Compare* S.O.S., Inc. v. Payday, Inc., 886 F.2d 1081, 1088 (9th Cir. 1989) (holding that federal policy requires interpretation of license against licensee in manner that withholds rights not expressly granted), *with* Bourne v. Walt Disney Co., 68 F.3d 621 (2d Cir. 1995) (applying traditional principles of contract interpretation to videotape licenses), *and* Uniform Computer Information Transactions Act § 307(f) (amended 2001) (stating the scope of a contract "must be construed under ordinary principles of contract interpretation").

41. As one discussion noted:

> Take all the approved Open Source Licenses and all the approved Free Software Licenses, and all of those not yet approved but which might be. Now arrange them from the simplest to the most complex. The GPL ranks up there next to those corporate legalese licenses. Now rank them according to their actual length. Ditto. Rank them according to the number of restrictions. Ditto.
>
> Frankly, the GPL is one of the most complex, lengthy and restrictive license [sic] you could

another.[42] The buggy state of these open source licenses is described below following a discussion of open source licensing principles.

1. *What makes a license "open source?"*

Hackers seem to agree on the basic objectives of open source licensing. There are four fundamental rights that an open source license needs to grant: First, access to source code; second, the right to run the software for any purpose; third, the right to change the software in any way; fourth, the right to redistribute the original software and any derivatives.[43] However, these fundamental principles are where the consensus begins and ends. In translating these basic objectives into an actual license contract, hackers have taken various paths. This should come as no surprise to anyone familiar with technology licensing. Drafting a license agreement requires the licensor to understand numerous fundamental issues.

For example, what is the scope of license? The hacker must craft the scope of a license so it enumerates what a licensee may do with the hacker's code (and what the licensee may *not* do). Who bears the risk of loss? The hacker must determine how much risk to bear, if any, in case the software fails to operate or operates in a manner that causes damage. What happens in case of copyright or patent infringement? The hacker must decide what happens if a licensee gets sued for intellectual property infringement due to the licensee's use of the hacker's code. These are fundamental issues in any software license and open source licenses are no different.

> devise and still manage to squeak into the "free" category. As a user, I am glad that GPL programs are free. As a developer, I wouldn't touch it with a ten foot lawyer.

Posting of Arandir to Ravicher, *Slashdot Discussion*, *supra* note 14, *see also* David Beckett, Inst. for Learning and Research Tech., Univ. of Bristol, Open Source-Closed License, *at* http://www.int.org/discovery/2001/02/licenses/ (last visited Feb. 2, 2002) (commenting that "understanding licenses is hard, and fuzzy" and that "[m]any licenses have been written by lawyers for lawyers and thus are basically impenetrable by most developers").

42. *See* Rick Moen, *A public discussion of open source licensing*, LINUX WORLD.COM, Sept. 21, 2002, *at* http://www.two-ld.com/man/2685/LWD00092licensing/ (last visited Oct. 20, 2002).

43. An expanded version of these fundamental principles is set out in *The Open Source Definition*, published by the Open Source Initiative, and *The Free Software Definition* published by the Free Software Foundation. *See* OPEN SOURCE INITIATIVE, *The Open Source Definition*, at http://www.opensource.org/docs/definition.php [hereinafter OSI, *The Open Source Definition*] (last visited Dec. 20, 2002); FSF, *The Free Software Definition*, *supra* note 31; *see also* Gomulkiewicz, *How Copyleft Uses License Rights*, *supra* note 7, at 185-93 (elaborating on the legal implications of open source principles).

An organization called the Open Source Initiative (OSI) has tried to bring some order to the open source licensing landscape.[44] OSI publishes criteria, called The Open Source Definition, which a license must meet in order to be considered "open source."[45] OSI also reviews licenses and declares whether a license meets The Open Source Definition.[46] If a license does fit The Open Source Definition, the licensor can claim that software provided under that license is OSI certified.[47] Companies such as Apple Computer, I.B.M., Intel, and Sun Microsystems have submitted licenses for evaluation by OSI.[48]

OSI's license review process is helpful but many complications remain. One complication is that a venerable hacker organization, the Free Software Foundation (FSF), offers a competing license evaluation formula. FSF determines whether a license is a "free software" license.[49] FSF's classification is based on whether the license fits both "the precise words" and the "spirit" of its free software definition.[50] According to FSF, free software licensing means something close to open source licensing but is not identical.[51]

2. Buggy open source licenses: General Public License and BSD-style License

Even if a programmer finds OSI's open source or FSF's free software designations useful, a programmer is still faced with the daunting task of choosing[52] from a wide assortment of potential license agreements.[53] Which

44. *See* OSI, *History of the OSI*, *supra* note 8.

45. *See* OPEN SOURCE INITIATIVE, *The Approved Licenses*, *at* http://www.opensouce.org/licenses/index.html [hereinafter OSI, *The Approved Licenses*] (last visited Dec. 20, 2002).

46. *Id.*

47. *Id.* OSI has published detailed procedures for obtaining certification. OPEN SOURCE INITIATIVE, *OSI Certification Mark and Program*, *at* http://www.opensource.org/docs/certification_mark.php (last visited Dec. 20, 2002).

48. OSI, *The Approved Licenses*, *supra* note 45.

49. Free software licensing includes the categories "copyleft" and "non-copyleft" licensing. See discussion of copyleft licensing, *infra* Part III.B.3.

50. FSF, *The Free Software Definition*, *supra* note 31.

51. *Id. See also* Open Source Initiative, *Why "Free Software" is better than "Open Source," at* http://www.gnu.org/philosophy/free-software-for-freedom.html (last visited Dec. 20, 2002).

52. A programmer could write his or her own license, and many do. However, OSI, FSF, and the leading hackers discourage this practice as not being in the best interest of the open source movement. *See* Moen, *supra* note 42 (discussing problems with license compatibility).

53. The OSI advises: "If you can, use one of the already-approved licenses for distributing your software. But be sure that you read and understand the license terms completely. We encourage you to select a license that is consistent with your business model." OSI, *The Approved Licenses*, *supra* note 45.

license best describes what the programmer wants others to do with his or her software?[54] For most programmers, no simple answer is available. For that reason, many programmers simply pick one of the two most common open source licenses, the General Public License (GPL)[55] or the Berkeley Software Distribution style[56] license (BSD), without careful analysis.[57] Unfortunately, these licenses are buggy. This article's purpose is not to point out every flaw, but to show that there are enough bugs to raise serious questions about the way in which open source licenses are maintained and improved.

3. The General Public License (GPL): Some of the Bugs

Richard Stallman of the FSF created the GPL[58] to propagate his vision of software programming freedom.[59] The GPL makes creative use of a contract

---

There are further issues as well. For example, are OSS licenses standard forms or more like treaties? Who can enforce and interpret OSS licenses?

54. *See id.*

55. "There are many OSS projects, and many licenses that govern them. If we look at them as a body, the GPL is the spine." Michael Tiemann, On 'Shared Source,' Speech given at the O'Reilly Open Source Convention (July 2001), *in* Open Source Initiative, *On 'Shared Source,' at* http://www.opensource.org/docs/sharedsource.html (last visited Dec. 20, 2002). The Free Software Foundation claims that "thousands of software projects" use the GPL. Press Release, Free Software Foundation, *The GNU General Public License Protects Software Freedoms* (May 4, 2001), *in* Free Software Foundation, *Press Releases*, *at* http://www.fsf.org/press/2001-05-04-GPL.html (last visited Dec. 20, 2002).

56. "BSD" stands for Berkeley Software Distribution, which was the name of the original UNIX-based distribution created at University of California, Berkeley. *See* Marshall McKusick, *Twenty Years of Berkley UNIX*, *in* OPENSOURCES, *supra* note 2, at 31, 33.

57. *See* Bezroukov, *The Catalog of Software Licenses*, *supra* note 11 ("[d]evelopers and firms who release products under free/open licenses usually don't understand the ramifications of the license, and generally don't spend much time thinking about the license. Often they make emotional decisions based on their perceptions of projects like FreeBSD, Linux, Perl, and Apache."); question from Dan to Mr. Ravicher. Ravicher, *Slashdot Discussion*, *supra* note 14 (asking for "boilerplate" to cover complex license transaction). The most common licenses are: the General Public License (GPL), The Lesser GPL, the Artistic License, the BSD-style license, and the Mozilla Public License. OSI lists 28 licenses on its "approved" list. *Id.* Some consider this diversity of license agreements to be, in software parlance, a feature rather than a bug. *See* Moen, *supra* note 42 (pointing to an audience member's comments that competition will breed better license agreements). In practice, though, most programmers do not have the time or inclination to wade through the choices or pay legal counsel to help them do it. When it comes to choosing a license, to paraphrase a popular song, programmers prefer freedom *from* choice to freedom *of* choice.

58. The FSF publishes a lengthy Frequently Asked Questions document on GPL-interpretation issues. Free Software Foundation, *Frequently Asked Questions about the GNU GPL*, *at* http://www.fsf.org/licenses/gpl-faq.html [hereinafter FSF, *FAQ*] (last visited Dec. 20, 2002).

59. *See* Ganesh Prasad, *The Practical Manager's Guide to Linux: Can you profitably use Linux in your organization?*, OSOPINION.COM, *at* http://www.osopinion.com/opinions/GaneshCPrasad/Ganesh

to reverse the copyright monopoly by permanently giving away the exclusive rights of a copyright holder,[60] what Stallman whimsically calls "copyleft."[61] The GPL tries to ensure[62] that any software licensed under it will be available in source code form for unfettered use, study, modification, and distribution.[63] One commentator noted that even though Mr. Stallman is a brilliant programmer, the contribution for which he most likely will be remembered is the GPL.[64]

The heart of the GPL is its license granting sections, which draw on key definitions.[65] The GPL covers three distinct scenarios: running a program; copying or distributing verbatim copies of a program; and modifying a program.

a. *Running a Program*

The GPL's treatment of the simple act of running a software program is enigmatic. On the one hand, the GPL says that "[a]ctivities other than copying, distribution and modification are *not covered* by this License; they are *outside of its scope*."[66] This statement seems to say that the GPL does not grant a license to run a program: running a program is "not covered." More disquieting is the statement that activities other than copying, distribution, and modification are "outside of its scope." Use of a copyrighted work outside the scope of a license is not only a breach of contract it is a violation of the Copyright Act.[67]

CPrasad2.html [hereinafter Prasad, *The Practical Manager's Guide*] (last visited Oct. 9, 2002).

60. A programmer copyrights a program once it is created and fixed in a tangible medium such as a computer hard drive. Copyright Act, 17 U.S.C. § 102 (2000). A copyright gives the programmer the exclusive right to reproduce, distribute, create derivative works, and publicly display the program. Copyright Act § 106. Then, the programmer offers to license the program under the terms of the GPL. Prasad, *The Practical Manager's Guide*, *supra* note 59. One of the terms of the GPL is that a licensee must get its sub-licensees to accept the same terms, and so on down the chain of distribution. *Id.* Another programmer accepts the license by exercising any of the software-owner's exclusive copyrights. *Id.*

61. Prasad, *The Practical Manager's Guide*, *supra* note 59.

62. *See* Ravicher, *supra* note 8 (discussing the enforceability of open source licenses, including the GPL).

63. FSF explains the goals of the GPL in its preamble. FSF, *GNU-GPL*, *supra* note 6, at pmbl.

64. Prasad, *The Practical Manager's Guide*, *supra* note 59.

65. FSF, *GNU-GPL*, *supra* note 6, §§ 0-4.

66. *Id.* § 0 (emphasis added).

67. S.O.S., Inc. v. Payday, 886 F.2d 1081, 1087 (9th Cir. 1989) ("A licensor infringes the owner's copyright if it exceeds the scope of its license."); Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994) (confirming the rational of *S.O.S. v. Payday*).

On the other hand, the GPL goes on to say that "[t]he act of running the Program is not restricted . . . ."[68] In one sense, this statement seems to answer "yes" to the question of whether running a program is permitted under the GPL. Confusion arises, however, because of the nature of licensing and the particular phraseology of the GPL. When anyone receives a copyrighted work, the first question he or she must ask is "what's permitted, if anything?"[69] In other words, the act of running a program may not be restricted, but one must ask whether it is explicitly permitted. The requisite words of permission do not appear in the permission-granting section of the GPL—running a program is not synonymous with copying, modification, or distribution, all of which are expressly permitted under the GPL.[70]

So, in the final analysis, do the authors of the GPL intend to allow a person to run a program? The answer is clearly "yes." The FSF definition says that the freedom to run a program, for any purpose, is one of the fundamental privileges of free software.[71] If running a program is permitted, why is the GPL ambiguous on the subject?

The FSF may believe that the GPL does not need to specifically grant a license to run a program because this right is already provided to owners of a copy under the "first sale" provisions and Section 117 of the Copyright Act.[72] While this rationale may or may not be technically accurate,[73] the GPL's wording[74] and approach unnecessarily cloud the issue.[75] If software comes with a written license, most users[76] expect the license document to describe

68. FSF, *GNU-GPL*, *supra* note 6, § 0.

69. The permission can be in the form of a copyright first sale (a type of limited license) or some other grant of rights. *See* Gomulkiewicz & Williamson, *supra* note 25, at 352-56.

70. By comparison, the X11 License says: "permission is hereby granted . . . to deal in the Software without restriction. . . ." X. Org., *X11 License*, *at* http://www.x.org/Downloads_terms.html [hereinafter *X11 License*] (last visited Dec. 20, 2002). The BSD License says that the rights it grants "are permitted," without mentioning restrictions at all. University of California, *BSD License*, *at* http://www.opensource.org/licenses/bsd-license.php [hereinafter *BSD License*] (last visited Dec. 20, 2002).

71. FSF, *The Free Software Definition*, *supra* note 31.

72. Copyright Act, 17 U.S.C. § 117 (2000).

73. It may not be a first sale when someone simply gives a copyrighted work for free with no explanation; the recipient may simply receive an implied license.

74. The ambiguity here might be just a matter of phrasing the sentence in the negative rather than the positive.

75. Fixing this bug in the GPL is simple. The GPL could say: "You have an unrestricted right to run the Program." The revised sentence should come before the "Activities other than copying . . ." sentence to clarify that the revised sentence is not modified by the "Activities other than copying . . ." sentence. *See* FSF, *GNU-GPL*, *supra* note 6, § 0.

76. This inattention to the end user perspective may reflect that fact that the primary audience for the GPL is software developers. The references to "you" in the GPL means each licensee of a GPL-licensed program. FSF, *GNU-GPL*, *supra* note 6, § 0. The GPL provides: "You may charge a fee for the physical

the full range of permitted uses. Users do not expect a written license to cover some usage scenarios and other uses to be covered by implication.[77] The GPL's approach would be more coherent if it simply granted the right to run the program or, at a minimum, pointed users to the source of the user's right to run the program.[78]

Another difficulty arises from FSF's view. A key tenet of open source licensing is that the hacker licenses the software without any warranties or responsibility for consequential damages.[79] If the GPL is not the contract that governs a user's right to run the software, Uniform Commercial Code Article 2's implied warranties may apply to the transaction and consequential damages would be available as a default rule.[80]

b. *Copying or Distributing Verbatim Copies*

(i) *License Grant*. In contrast to the GPL's enigmatic treatment of the right to run a program, the GPL's license to copy and distribute verbatim copies of source code is relatively straightforward. Section 1 of the GPL permits the user to "copy and distribute verbatim copies of the Program's source code as [he or she] receive[d] it."[81] This license is conditioned on three requirements: the copier or distributor must [1] "conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; [2] keep intact all notices that refer to [the GPL] and to the absence of any warranty; and [3] give any other recipients of the Program a copy of [the GPL] along with the Program."[82]

(ii) *Fees*. Following the license grant and conditions, the GPL contains a paragraph that addresses the issue of charging fees for a GPL-licensed

act of transferring a copy, you may, at your option offer warranty protection in exchange for a fee." *Id.* at § 1. These references to "you" make the most sense when directed to a software developer.

77. Licenses, if written properly, give users detailed information about what they can and cannot do with software. *See* Gomulkiewicz & Williamson, *supra* note 25, at 346-52 (explaining the educational benefits of mass market licenses).

78. For instance, the GPL could say: "The GPL does not describe your right to run the Program; copyright law gives you an unrestricted right to run the Program."

79. *See* FSF, *GNU-GPL*, *supra* note 6, §§ 11-12 (disclaiming warranties); Gomulkiewicz, *How Copyleft Uses License Rights*, *supra* note 7, at 191-93 (describing how risk-shifting is fundamental to open source licensing).

80. *See* Robert B. Mitchell, *Software and Data Transactions Under Article 2 of the U.C.C.*, DATALAW REP., Sept. 1995, at 14, 25-26; *but see* Lorin Brennan, *Why Article 2 Cannot Apply to Software Transactions*, 38 DUQ. L. REV. 459 (2000).

81. FSF, *GNU-GPL*, *supra* note 6, § 1.

82. *Id.*

software program.[83] Since this paragraph follows the license grant and prerequisites, it presumably describes the right to charge money for exercising the rights granted immediately before it. The GPL states that a licensee may "charge a fee for the physical act of transferring a copy" and that the licensee "may at [the licensee's] option offer warranty protection in exchange for a fee."[84] This phrasing raises several issues.

One issue is whether these statements about fee charging are additional license conditions, a separate covenant, or merely advisory. Determining the proper classification has serious ramifications.[85] If a licensee charges a fee in violation of the GPL, the nature of the consequence for the violation will depend upon the nature of the obligation undertaken. If the statements were advisory, there would be no consequence for ignoring them; if they were a covenant, there would be a breach of contract; if they were a license condition, there would be a copyright infringement.

The most reasonable conclusion is that the wording is advisory.[86] The FSF is constantly explaining that "free software" refers to freedom of use, not price.[87] The fee-charging language in Section 1 of the GPL was likely drafted to clarify that point. An explanation of this nature would not be out of place in the GPL. Several parts of the GPL appear to contain advice or explanations[88] rather than covenants. In addition, where the GPL addresses fee charging in a later license grant, it clearly says that the licensee may charge no fees whatsoever.[89]

However, if the fee-charging language in Section 1 is a covenant or license condition, a licensee must understand what fees the GPL allows. Copyright law, which underlies software licenses, colors the meaning of the fee-charging wording.[90] The right to make copies and the right to distribute

83. *Id.*

84. *Id.*

85. *See* Sun Microsystems, Inc. v. Microsoft Corp., 188 F.2d 1115, 1122 (9th Cir. 1999).

86. The GPL could have been clear about its intentions, one way or the other, if it had said either "[t]hese are the only fees you can charge for a Program" or "[y]ou can also charge other fees for the Program except as explained in Section 2."

87. FSF, *The Free Software Definition*, *supra* note 31.

88. *See*, *e.g.*, FSF, *GNU-GPL*, *supra* note 6, § 7 para. 3, 4 (explaining how § 7 relates to the rest of the GPL); *id.* § 10 (describing the appropriate process for combining software with software licensed under inconsistent license terms).

89. *Id.* § 2(b). Section 2 of the GPL says that the licensee may exercise the license rights only at "no charge" to third parties. *Id.* By contrast, the license grant in Section 1 does not explicitly forbid the charging of fees to third parties. *Id.* § 1. Comparing the text of one section of a contract to understand the text of another section is classic contract interpretation analysis.

90. *See* Raymond T. Nimmer, *Breaking Barriers: The Relations Between Contract and Intellectual*

copies are separate and distinct rights under copyright law.[91] The GPL, in Section 1, grants the right to make verbatim copies as well as distribute verbatim copies.[92] However, while Section 1 permits a licensee to charge a fee for the act of distributing verbatim copies, the GPL does not mention whether a licensee can charge a fee for the act of making verbatim copies.[93]

c. *Modifying a Program*

GPL Section 2's complex license grant and the conditions that follow it[94] have vexed many readers of the GPL. One troubling aspect of the license grant[95] is ascertaining whether it grants a license only to modify software or whether it grants a broader license to create any type of derivative work. Another concern arises from the breadth of the license's provision that licensees to provide their derivative works and the corresponding source code to anyone who wants it at no charge.[96]

(i) *Scope of License*. The license grant in Section 2 provides that: "You may modify your copy or any copies of the Program or any portion of it, *thus forming a work based on the Program* . . . ."[97] Does this provision grant only

Property Law, 13 BERKELEY TECH. L.J. 827, 860-88 (1998) (discussing the effects of intellectual property laws on intellectual property contracts).

91. *Compare* Copyright Act, 17 U.S.C. § 106(1) (2000) (granting an owner exclusive rights to make copies), *with* Copyright Act § 106(3) (granting an owner exclusive rights to distribute copies of the work).

92. FSF, *GNU-GPL*, *supra* note 6, § 1.

93. Section 4 of the GPL also distinguishes between copying and distribution. *See id.* § 4 ("**You may not copy**, modify, sublicense, or **distribute** the Program except as expressly provided under this License." (emphasis added)). It is possible that the authors of the GPL did not believe that anyone would ever want to charge a fee merely for making a copy, or they may have thought that making a copy is part and parcel with distributing a copy. On the other hand, while FSF states emphatically that anyone should be able to sell copies, it does not clearly say that anyone should be able to charge a fee for making a copy. *See* Free Software Foundation, *Selling Free Software*, *at* http://www.fsf.org/philosophy/selling.html (last visited Dec. 20, 2002); FSF, *The Free Software Definition*, *supra* note 31; Free Software Foundation, *Categories of Free and Non-Free Software*, *at* http://www.fsf.org/philosophy/categories.html (last visited Dec. 20, 2002).

94. *See* Jeff C. Dodd & Brian Martin, *Building A Cathedral Over the Bazaar*, DOING BUSINESS ONLINE (2000); Wendy C. Freedman, *Open-source vies with classic IP model*, NAT. L.J., Mar. 13, 2000, at B14; MICROSOFT LICENSING, MICROSOFT CORP., *Some Questions Every Business Should Ask About the GNU General Public License (GPL)* (downloadable document under General Public License Analysis Frequent Asked Questions), *at* http://www.microsoft.com/resources/sharedsource/default.mspx [hereinafter MICROSOFT LICENSING, MICROSOFT CORP., *Questions About the GNU-GPL*] (last visited Dec. 20, 2002).

95. FSF, *GNU-GPL*, *supra* note 6, § 2, para. 1.

96. *Id.* § 2(b) ("You [the licensee] must cause any work that you contribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this license.").

97. *Id.* § 2 (emphasis added).

the right to modify a program, or a broader right to create any derivative work?[98] Several reasons suggest that the narrower grant is more plausible.[99] Section 2's "thus creating" seems to merely remind the reader that a modification is a type of work[100] based on a program.[101] The GPL defines the term "work based on a Program" to mean "either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language."[102] In addition, two of the license conditions in Section 2 refer to "the modified program" and "the modified file."[103] Similarly, the language that appears in the three paragraphs following Section 2's license conditions also refers to "the modified work."[104]

However, some language supports the interpretation that Section 2 grants a broad right to make derivative works. The part of the license grant that describes distribution rights says that the licensee has the right to distribute "such modifications *or* work,"[105] suggesting that the license grants the right to create both a modification and a work based on the program. Also, the license condition in Section 2(b) applies to more than modifications of the Program. It also applies to a program that "contains" or is "derived from" a licensed program.[106] Further, the GPL's Preamble suggests that the GPL grants a broad license to create derivative works. The Preamble says that the GPL was designed so that programmers can "change[107] the software *or* use pieces of it in new programs."[108]

(ii) *Scope of 2(b) License Condition.* Section 2(b) is at the heart of copyleft. This section requires a developer to publish his or her source code

98. One would expect the license grant to say something like: "You may create a work based on a Program or any portion of it."

99. *See* Copyright Act, 17 U.S.C. § 101 (defining derivative work, in part, as one which includes "modifications" consisting of "original works of authorship").

100. Section 2 of the GPL is saying that a modified program is a work based on a Program. However, the definition of a "work based on a Program" already encompasses modified programs—a modified work is a subset of all derivative works. *Id.*

101. Under this language, for instance, simply combining two programs would create a work based on the program, even though neither program is modified in the process.

102. FSF, *GNU-GPL*, *supra* note 6, § 0.

103. *Id.*

104. *Id.*

105. *Id.* § 2 para. 1 (emphasis added). The reference to "work" presumably refers to the defined term "work based on the Program."

106. *Id.* § 2(b).

107. In other words, "modify."

108. *Id.* at pmbl.

for free use[109]—and in this case (with apologies to Mr. Stallman) "free" refers to both free speech and free beer.[110] This section also makes the GPL a "viral" license because it allows one developer to impose GPL license terms on another developer's code. Section 2(b) says that a licensee "must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this license."[111]

What is the breadth of the condition? Section 2(b)'s conditions apply to works "derived from" other works and works that "contain" other works. A work that contains another work is usually characterized as a derivative work.[112] Does the GPL's use of both words mean that it treats "derived works" as something different and potentially broader than copyright derivative works? Does "derived from" mean derived from non-copyrightable aspects of the program, such as ideas or data? If so, Section 2(b)'s license condition may apply to programs derived from minuscule amounts of code or non-copyrightable code that would not otherwise make the host program a derivative work according to copyright law. It is possible that the GPL merely seeks to clarify and emphasize that this condition covers certain types of derivative works, namely works contained in other works. Unfortunately, the intent of the GPL is murky.

Following the license grant and conditions in GPL Section 2, three paragraphs (Explanatory Paragraphs) attempt to clarify the license grant and conditions,[113] especially the critical distinction in Section 2(b) between derivative and collective works on the one hand and separate and independent works on the other. Reading these paragraphs is like attempting to tune in a distant radio station—moments of clarity followed by moments of distortion. The Explanatory Paragraphs say that if a programmer creates a modified work based on the licensed program and if identifiable parts of that modified work are not derived from the licensed program and if those parts of the modified work can be "reasonably considered independent and separate," and if those parts of the modified work are distributed as a separate work that would not qualify as a derivative work of the licensed Program, then the GPL does not apply to those parts of the modified work in that context.[114]

109. *Id.* at pmbl. (stating the GPL is intended "to make sure the software is free for all its users.").
110. See *supra* note 31.
111. FSF, *GNU-GPL*, *supra* note 6, § 2(b).
112. *See* Copyright Act, 17 U.S.C. § 101 (2000) (defining derivative work).
113. *See* FSF, *GNU-GPL*, *supra* note 6, § 2(a)-(c).
114. *See id.*

Attempting to clarify this provision further, the GPL states: "it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program."[115] It also states: "mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of storage or distribution medium does not bring the other work under the scope of this License."[116]

The Explanatory Paragraphs are relatively helpful—at least as helpful as they can be under the circumstances. They track the intent of the Copyright Act closely.[117] Even so, the distinction they try to make is complex and extremely fact intensive in the software context. FSF's GPL Question & Answer document illustrates this well when it concludes that often the distinction comes down to how "intimate" two programs are with one another.[118] The difficulty comes less from ill chosen words than from making such serious consequences turn on a distinction that is inherently uncertain.[119]

Even if one could draw the line with certainty serious issues remain. Software engineering practices have evolved significantly since the GPL was last revised.[120] Software has become more complex. This bears profoundly on the interpretation of the terms "independent work" and "separate work," which are terms that establish whether a work is subject to GPL Section 2(b). Use of object oriented programming and remote procedure calls make the concepts of independent and separate work more difficult to define. Technology is becoming more connected not more separate.[121]

---

115. *Id.* § 2.

116. *Id.*

117. *See* Copyright Act §§ 101-122.

118. *See* FSF, *FAQ*, *supra* note 58 (considering the question of what makes a mere aggregation).

119. The consequence is that a programmer's code becomes freely available to anyone free of charge whether the programmer desires that result or not. It is one thing to freely volunteer for the free software army, but forced conscription is another matter.

120. The current version of the GPL (Version 2) dates from 1991. FSF, *GNU-GPL*, *supra* note 6.

121. One of the most troublesome aspects of the GPL is its failure to account clearly for linking and other forms of communication between applications software and systems software. FSF seems to take a broad view of the types of communication that trigger the copyleft aspects of the GPL. *See* discussion *supra* Part III.B.2. Contrast FSF's approach, however, to that of Linus Torvalds. Mr. Torvalds takes the position that making systems calls to the Linux kernel does not trigger the copyleft aspects of the GPL. Mr. Torvalds' clarifying note to the GPL as applied to Linux says:

> NOTE! This copyright does not cover user programs that use kernel services by normal system calls—this is merely considered normal use of the kernel, and does not fall under the heading of "derived work." Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the Linux Kernel) is copyrighted by me and others who actually

In addition, programs that were "reasonably considered independent" are no longer so considered. Directories, TCP/IP stacks, fonts, spell checkers, clip art, file systems, compression software, and perhaps even operating system kernels[122] may or may not be independent works, depending upon how they are packaged or configured. Certain issues specific to the development of embedded software complicate the distinction between separate and combined works.[123] Interpreted programs, such as those written in Java, present unique GPL interpretation issues.[124] As one commentator noted: "In an era of category-blurring inventions such as document macro-procedures, how are license-covered *programs* to be distinguished from non-licensed configurations and data *files*?"[125]

4. *The BSD License: Some of the Bugs*

The BSD License is so named because it is the end user license that accompanied the Berkeley Software Distribution of UNIX.[126] The BSD License is constantly compared to the GPL.[127] However, the look and feel of the two licenses differs dramatically. The GPL reads like a combination of a political tract, philosophy dissertation, how-to guide for non-lawyers, and lengthy, complex license contract. The BSD License, by contrast, is short and unremarkable, appearing in content and in form much like many other mass market licenses.

---

wrote it. Linus Torvalds.

Linus Torvalds, *Introduction to Linux GNU Kernel Public License*, *at* http://www.linux.de/linux/gnu.html [hereinafter Torvalds, *Introduction to Linux-KPL*] (last visited Dec. 20, 2002).

122. Linux is the kernel in various operating system packages, such as Red Hat Linux, GNU/Linux, and Debian GNU/Linux. *See id.* The Mach kernel is another example of an operating system kernel on which several operating systems are based.

123. *See* Farhad Manjoo, *Open Source's Dot-Net Less Open*, WIREDNEWS, Jan. 28, 2002), *at* http://wired.com/news/print/0,1294,50037,00.html (last visited Dec. 20, 2002) (discussing problems with the GPL in modern software development). Embedded software developers Wind River and Caldera have expressed concern with the uncertain viral effects of the GPL. MICROSOFT LICENSING, MICROSOFT CORP., *Questions About the GNU-GPL*, *supra* note 94.

124. *See* FSF, *FAQ*, *supra* note 58 (discussing specific Java related questions).

125. *See* Moen, *supra* note 42 (emphasis in original).

126. *See* Marshall McKusick, *Twenty Years of Berkley UNIX: From AT&T-Owned to Freely Redistributable*, *in* OPENSOURCES, *supra* note 2, at 31-46 (discussing the history of Berkeley Unix). There are 3 primary open source versions of BSD UNIX. NetBSD focuses on multi-platform support, FreeBSD focuses on the PC architecture, and OpenBSD focuses on security. *Id.* at 43.

127. *See* PETER WAYNER, FREE FOR ALL 77-103 (Harper-Collins 2000) (comparing the evolutions of the GPL and BSD License).

Aside from looks, the two licenses also present very different licensing philosophies. Hackers who prefer the BSD License say that it provides the user more freedom than the GPL.[128] Under the BSD License, a hacker can do anything he or she wants to do with the licensed software.[129] A hacker has the freedom to make derivatives that are open or closed source,[130] to share or not share modifications, and to charge or not charge fees of any nature.[131]

The BSD License grants the following license: "Redistribution and use in source and binary forms, with or without modification, are permitted" provided that certain conditions are met.[132] The license contains three conditions: (1) if the licensee redistributes the software in source code form, the licensee must retain any copyright notices, and inform the downstream licensees of all the conditions and the disclaimers; (2) if the licensee redistributes the software in binary form, the licensee must reproduce any copyright notices, and inform the downstream licensees of all the conditions and the disclaimers in the documentation and/or other materials that accompany the software; (3) the name of the licensor may not be used to promote or endorse any product derived from the licensed code.[133]

Bugs can easily be found in the colorful and complex GPL, but the staid and simple BSD License has bugs too,[134] including the three described below.

The license grant could be improved. Most license grants tell the licensee which copyrights are being licensed.[135] The BSD License grants the rights of "use" and "redistribution."[136] "Use"[137] is not one of the exclusive rights of a

---

128. *See id.* at 97 ("The debate between BSD-style freedom and GNU-style freedom is one of the greatest in the free programming world and is bound to continue for a long time as programmers join sides and experiment.").

129. *BSD License*, *supra* note 70.

130. *See* Stephen Shankland, *Ximian changes open-source license*, CNETNEWS.COM, Jan. 27, 2002, *at* http://news.com.com/2100-1001-823734.html (last visited Dec. 20, 2002) (discussing the advent of a BSD-style pen-source license which permits full use in closed-source projects).

131. *BSD License*, *supra* note 70. (The GPL requires anyone who makes a derivative to share code back to the community.) *Compare* FSF, *GNU-GPL*, *supra* note 6, § 2(b), *with BSD License*, *supra* note 70.

132. *BSD License*, *supra* note 70.

133. *Id.*

134. The amount of attention that Richard Stallman gave to drafting the GPL stands in contrast to the lack of attention paid to the choice of license for BSD UNIX. Reportedly, Bill Joy, the father of BSD UNIX "just copied over a license from the University of Toronto . . . He simply wanted to get the source code out the door." *See* WAYNER, *supra* note 127, at 94.

135. *See* Gomulkiewicz & Williamson, *supra* note 25, at 346-56.

136. *BSD License*, *supra* note 70.

137. Granting a right to "use" is common practice, however, though the licensor usually describes what particular uses it is permitting. *See* Nat'l Car Rental Sys. Ins. v. Computer Assoc. Intl., Inc., 991 F.2d 426, 431-32 (8th Cir. 1993) (noting that courts have held contractual limits on computer program use are

copyright owner,[138] although it is one of the exclusive rights of a patentee;[139] "distribution" is an exclusive copyright right,[140] but the BSD License only mentions redistribution.[141] Moreover, although the license mentions that the licensee may use or redistribute the licensed software with or without modifications, the license never affirmatively grants the right to modify.[142]

The warranty disclaimer and limitation of liability section could also be improved. Many licensors recognize that a court could refuse to enforce their disclaimer or limitation in certain contexts, such as consumer transactions.[143] To plan for that contingency, it is common practice for a licensor to deploy a severability clause or the qualification "to the extent permitted by applicable law."[144] The BSD License does not contain either contingency even though shifting risk away from the hacker is a fundamental tenet of open source development.[145]

When the BSD License was created, few, if any, courts had addressed the enforceability of mass market licenses. Today, most courts uphold the enforceability of mass market licenses,[146] but only when the licensor presents the potential licensee with an opportunity to review the license and the licensee manifests assent to it.[147] The BSD License informs the licensee how

distinct from copyright rights). If the licensor intends to permit any use whatsoever, the license grant should say something like: "You may use the software in any way you want."

138. The exclusive rights are the right to reproduce, distribute, make derivative works, and publicly perform or display. *See* Copyright Act, 17 U.S.C. § 106 (2000).

139. Patent Act, 135 U.S.C. § 271 (2000) (categorizing infringement as use without authority).

140. Copyright Act § 106(3).

141. *BSD License*, *supra* note 70.

142. *See id.* A modification is a type of derivative work. Copyright Act § 101.

143. See Gomulkiewicz, *The License Is the Product*, *supra* note 33, at 905-07 (discussing the challenges of drafting proposed U.C.C. Article 2B in light of the conflicts on warranties between software industry practice and U.C.C. Article 2 default rules). *See generally* James J. White, *Default Rules in Sales and the Myth of Contracting Out*, 48 LOYOLA L. REV. 53 (2002) (describing the mismatch between industry practice and U.C.C. Article 2 liability and remedy default rules). White proposes that the default rules should be reconsidered. *Id.*

144. *See*, *e.g.*, FSF, *GNU-GPL, supra* note 6, § 11. A severability clause allows a court to strike one provision from a contract while leaving the other provisions in place.

145. *See* FSF, *GNU-GPL*, *supra* note 6, at pmbl. (clarifying that in open source licensing a programer is not liable under any warranties if a hacker modifies the Program). As Bruce Perens explains, "[i]f free-software authors lose the right to disclaim all warranties and find themselves getting sued over all the performance of the programs that they've written, they'll stop contributing free software to the world." Perens, *The Open Source Definition*, *supra* note 2, at 181. *See also* Gomulkiewicz, *How Copyleft Uses License Rights*, *supra* note 7, at 191 (discussing the fundamental need for risk shifting in copyleft).

146. *See* Ravicher, *supra* note 8, ¶¶ 49-52; Gomulkiewicz, *supra* note 20, at 450-51 (citing cases). The seminal case is ProCD v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996). *See also* Bowers v. Baystate Tech., 302 F.3d 1334 (Fed. Cir. 2002).

147. *See* Specht v. Netscape, 306 F.3d 17, 23, 29-30 (2d Cir. 2002) (party not bound by terms of

it must present copyright notices and disclaimers to further licensees.[148] However, it contains no such instructions to the licensee[149] on how it should present the license to further licensees to maximize enforceability.

C. The Significance of Buggy Open Source Licenses

Software bugs come in many levels of severity. Some bugs are cosmetic or relatively insignificant. Other bugs are troublesome but only arise in certain contexts. Others jeopardize the functionality of important features. Still others are called showstoppers because they threaten to shut down everything.[150] Despite the range of software bug severity, the goal of the open source development process is to fix the bugs and the question is usually not "if" but "when."[151]

Similarly, the bugs in open source licenses range from trivial to very severe. Yet, while open source developers pride themselves in the number of eyeballs devoted to fixing buggy software, buggy open source licenses suffer from inattention. To make matters worse, the authors of the most important open source licenses do not regularly evaluate the licenses and fix them when they are broken.

Hackers know that effective licenses are essential to open source development. As one hacker group puts it: "To stay free, software must be

---

license without reasonable opportunity to review and manifestation of assent); Gomulkiewicz, *The License Is the Product*, *supra* note 33, at 895-904. The Uniform Computer Information Transactions Act (UCITA) also requires an opportunity to review and manifestation of assent. Uniform Computer Information Transactions Act §§ 112, 209 (amended 2001).

148. *BSD License*, *supra* note 70.

149. *Compare id.*, with FSF, *GNU-GPL*, *supra* note 6, § 5. Arguably, the BSD License could be improved by adding other terms, such as a term addressing when or under what circumstances the license may be terminated.

150. *See generally* G. PASCAL ZACHARY, SHOW-STOPPER!: THE BREAKNECK RACE TO CREATE WINDOWS NT AND THE NEXT GENERATION AT MICROSOFT (1994).

151. *See* Raymond, *How To Become a Hacker*, *supra* note 9 ("In this imperfect world, we will inevitably spend most of our software development time in the debugging phase."); Robert Lemos, *Torvalds, developers at odds over Linux*, CNETNEWS.COM (Jan. 30, 2002), *at* http://news.com.com/2100-1001-826093.html [hereinafter Lemos, *Torvalds*] (last visited Dec. 20, 2002) (explaining that even if an open source development project has many eyeballs catching bugs and proposing fixes, the chief architect of the project may be too slow to incorporate those fixes, thus becoming a bottleneck); Robert Lemos, *Site to pool scrutiny of Linux security*, CNET NEWS.COM, Jan. 6, 2002, *at* http://news.com.com/2100-1001-830130.html (last visited Dec. 20, 2002) (explaining that although the many eyeballs approach has the potential to make open source code very secure, that promise is not being fulfilled at present).

copyrighted and licensed."[152] Buggy licenses impede the success of open source software, but well crafted licenses could have the opposite effect.

V. A Standards Organization For Open Source Licenses

Most hackers want fewer and better licenses from which to choose. Despite the entreaties of open source notables to use existing license forms, hackers keep creating new forms because the old ones are buggy and do not meet their licensing objectives. The best efforts of the OSI and FSF have not solved the problem. An open source license organization (OSLO) could provide an important service to developers and users of open source software by creating, maintaining, and improving key open source licenses.

This section begins by describing the two primary types of standards, and the role that standards organizations play in the information technology industry.[153] It then explains the difference between open standards organizations and open source development communities. It concludes by discussing the type of standard that the GPL and BSD License have become, and why a different type of standard is preferable.

A. Role of Standards Organizations

Technological standards originate in one of two ways. Some standards arise from the widespread adoption of a given technology. These are sometimes called *de facto* standards.[154] An example of a *de facto* standard is the Graphics Interchange Format (GIF) approach to compressing graphical images.[155] Other standards arise when people from the industry come together to collaborate on a technology.[156] These standards are known as *de jure* standards. The XML information interchange standard is an example of a *de jure* standard.[157]

152. Debian GNU/Linux, *What Does Free Mean?*, *supra* note 6.

153. *See generally* Mark A. Lemley, *Intellectual Property Policy and Standard-Setting Organizations*, 90 Cal. L. Rev. 1889 (2002).

154. NYS Forum for Information Resource Management, Rockefeller Institute of Government, *Information technology standards: who decides and why*, OPEN FORUM, Jan. 1996, at 1, 1 [hereinafter Information Resource Management, *Information technology standards*].

155. *Id.*

156. *Id.*

157. *See* Wylie Wong, *XML as the great peacemaker*, CNET NEWS.COM, Dec. 21, 2000, *at* http://news.com.com/2100-1001-250205.html (last visited Dec. 20, 2002).

Information technology, in its initial stages, consisted of stand alone systems developed by individual vendors. Over time, networking these systems became important. Consequently, industry members began to come together in standards organizations to create technical specifications aimed at achieving integration.[158] These bodies[159] and the *de jure* standards they create now play an important role in integrating disparate systems, including the vast computer network we call the Internet.[160] Prominent Internet standards organizations include the Internet Society (ISOC),[161] World Wide Web Consortium (W3C),[162] and the Internet Engineering Task Force (IETF).[163] At a mundane level, these organizations create, maintain, and update technical specifications; at a higher level, they liberalize trade by removing technical barriers.[164]

Some people confuse open source development with the development of *de jure* standards by bodies such as the ISOC, W3C, and IETF. They are not the same thing.[165] Open standards bodies, by their very nature, have formal organizations and processes.[166] Open source projects arise informally and

---

158. *See*, *e.g.*, World Wide Web Consortium, . . . *in 7 points*, *at* http://www.w3c.org/consortium/points/ (last visited Dec. 20, 2002).

159. Standards organizations differ in the way they are organized. Some organizations, such as the International Standards Organization (ISO), have limited membership but permit relatively widespread public input. Others, such as the Internet Engineering Task Force (IETF), encourage public input and participation. *See* Internet Engineering Task Force, *Participating in the efforts of IETF*, *at* http://www.ietf.org/join.html (last visited Dec. 20, 2002) ("The IETF is not a member organization (no cards, no dues, no secret handshakes :-)."). Standards organizations also differ in how quickly they can act. Some are organized to act quickly to respond to industry change. Others tend to act slowly and deliberately, favoring formality and political process over speed. *See* Paul Festa, *Critics clamor for Web services standards*, CNETNEWS.COM, Feb. 12, 2002, *at* http://news.com.com/2100-1023-834990.html (last visited Dec. 20, 2002).

160. *See* Barry M. Leiner et al., *A Brief History of the Internet*, INTERNET SOCIETY, *at* http://www.isoc.org/internet/history/brief.shtml (last visited Dec. 20, 2002).

161. ISOC is the organizational home of the Internet Engineering Task Force, the Internet Architecture Board, the Internet Engineering Steering Group, and the Internet Research Task Force. *See* Internet Society, *All About the Internet Society*, *at* http://www.isoc.org/standards/ (last visited Dec. 20, 2002).

162. *See* World Wide Web Consortium, *About the World Wide Web Consortium*, *at* http://www.w3.org/Consortium/ [hereinafter W3C, *About W3C*] (last visited Dec. 20, 2002).

163. *See* Internet Engineering Task Force, *Overview of the IETF at* http://www.ietf.cnri.reston.va.us/overview.html (last visited Dec. 20, 2002).

164. *See* ECMA, *ECMA—Standardizing Information and Communication Systems*, *at* http://www.ecma-international.org/MEMENTO/PREFACE.HTM (last visited Dec. 20, 2002); Scott Bradner, *The Internet Engineering Task Force*, *in* OPENSOURCES, *supra* note 2, at 47.

165. Even though open standards are different than open source software programs, open standards often benefit open source development. *See* Bradner, *supra* note 164, at 52.

166. *See*, *e.g.*, International Standards Organization, *ISO Structure*, *at* http://www.iso.ch/iso/en/

spontaneously when a programmer sees something that needs to be built or fixed—as Eric Raymond likes to say, an itch that needs to be scratched.[167] An open source project is born when the developer posts his or her code on the Internet, and it grows into a community as the code attracts developers who have the same itch to scratch.[168] The community of developers provides outside input to the originator much like the members of a standards community. However, in the end, the originator of an open source development is the lord of the project. He or she makes the final decisions about how the code is built and fixed.[169] If the members of the community become dissatisfied, their primary recourse is to start a competing project, what hackers call forking the code base.[170]

These open source development communities do not create *de jure* standards, but they sometimes create *de facto* standards. Many important technologies underlying the Internet are open source developments. For example, the Apache web server and the Sendmail email transport agent are Internet standards.[171] The Perl software language is used to develop many popular websites.[172] Linux based operating systems not only are common on web servers, they are quickly becoming the *de facto* standard for UNIX based

aboutiso/isostructure/isostr.html (last visited Dec. 20, 2002).

167. Raymond, *The Cathedral and the Bazaar*, *supra* note 1.

168. Linus Torvalds' Linux development project is the most famous example of this model. In other cases, though, several developers who have the same idea come together at the same time to create an open source development. The Apache web server development project is the most famous example of this model. *See* The Apache Software Foundation, *Welcome*, *at* http://www.apache.org (last visited Dec. 20, 2002).

169. *See*, *e.g.*, Lemos, *Torvalds*, *supra* note 151 (discussing Torvalds' control over the Linux open source project).

170. *See* Eric S. Raymond, *Homesteading the Noosphere*, *at* http://www.tuxedo.org/~esr/writings/homesteading/homesteading/index.html [hereinafter Raymond, *Homesteading the Noosphere*] (last visited Sept. 22, 2002) (describing forking as the process where programmers begin with a single program and develop it in different directions); WAYNER, *supra* note 127, at 207-38 (discussing forking and its history and value in open source programming).

171. *See* Tim O'Reilly, *Open Source: The Model for Collaboration in the Age of the Internet*, keynote speech given at Computers, Freedom and Privacy (Apr. 6, 2002), *at* http://www.oreillynet.com/pub/a/network/2000/04/13/CFPkeynote.html (last visited Dec. 20, 2002) ("It is questionable whether the web would still be open without Apache, or e-mail without Sendmail."); Josh McHugh, *For the Love of Hacking*, FORBES, Aug. 21, 1998, at 94, 99-100 (noting Sendmail routes 80% of the internet's email and Apache serves more than 50% of the internet's websites); O'Reilly, *The Open Source Revolution*, *supra* note 31 (describing Sendmail as "the backbone of the Internet's email server infrastructure" and Apache as having "53 percent of all visible web servers").

172. O'Reilly, *The Open Source Revolution*, *supra* note 31 (asserting the "Perl language is the undisputed king of the open-source programming languages").

operating systems and one of the most popular operating systems of any type.[173]

A *de facto* standard only remains the standard, however, so long as it is the best overall solution. Over time, a *de facto* standard that provides a better solution may replace the reigning *de facto* standard.[174] In other cases, the industry may come together to create a *de jure* standard in place of a *de facto* standard that is inadequate.[175]

In the same way the industry adopts technology standards, it adopts other types of standards as well. The licenses that accompany open source code have become the *de facto* standards for open source licensing. The GPL has become the standard copyleft open source license and the BSD License has become the standard non-copyleft open source license. As described in this article, these licenses need mending or even replacing.

The authors of the GPL and the BSD License could simply improve these licenses, but this is easier said than done. Despite a constant stream of complaints, FSF has not changed the GPL in over a decade. FSF's inaction suggests that it does not have the motivation to fix known problems with the GPL on a regular basis. Fixes to the BSD License have also been slow to come.

Even if the authors try to fix the licenses they cannot guarantee that the fixes will be satisfactory or comprehensive, or that the fixes will take into account the opinions of the broad constituencies that now use and interact with open source software. More importantly, no guarantee can be given that the licenses will be fixed regularly in the future. Open source developers and their lawyers should come together to fix the buggy *de facto* standard open source licenses so that open source software has high quality licenses that work well today and into the future.

173. *See Week in review: In Linux land*, ZDNET NEWS, Feb. 1, 2002, *at* http://zdnet.com.com/2100-11-828055.html (last visited Dec. 20, 2002); Linus Torvalds, *The Linux Edge*, *in* OPENSOURCES, *supra* note 2, at 101.

174. *See* GATES, *supra* note 21, at 35-64 (discussing how products can become obsolete unless they establish trends rather than follow them).

175. *See*, *e.g.*, Matthew Broersma, *Embedded Linux crying out for standards*, ZDNET NEWS, May 17, 2002, *at* http://zdnet.com.com/2100-1104-916331.html (last visited Dec. 20, 2002) (discussing the compatibility challenges caused by the open source process that has resulted in different Linux distributions). Good historical examples include the ANSI C++ and POSIX developments.

B. Open Source License Organization (OSLO)

The general principles for open source licensing are reasonably well developed.[176] Nonetheless, the licenses that implement these principles could be improved greatly by creating an open source license organization (OSLO) dedicated to the improvement and maintenance of open source licenses. OSLO could not only fix the license bugs we know exist today, it could also regularly fix new bugs as they are discovered. It could create new and better license forms to implement additional open source licensing models. OSLO's licenses would have the benefit of input from the broad constituencies that now use and interact with open source software. The work that OSLO might do and the form it might take are described in detail below.

1. OSLO's Work

OSLO could first create improved versions of the GPL and the BSD License that are true to the objectives of these licenses. In the case of the GPL, the task would not be to simply tidy up the license—FSF permits verbatim copying of the GPL but no modifications.[177] This restriction is probably for the best anyway, given the number and severity of GPL bugs. Regardless, it is always good practice for licensors to reexamine their end user licenses from time to time to see if the writing can be improved.[178] The BSD License could be improved by revising and updating the language to better account for intellectual property law and recent developments in commercial law. OSLO's BSD-style license could take the place of a myriad of similar but slightly different licenses such as the Apache license and the X11 license.[179]

OSLO could take on other projects as well. It could create licenses for licensing scenarios that the GPL and BSD License models do not address.[180]

176. *See supra* Part III.B.1.

177. FSF, *GNU-GPL*, *supra* note 6, at pmbl.

178. *See* Gomulkiewicz & Williamson, *supra* note 25, at 366 (challenging lawyers to improve license readability). *Compare*, *e.g.*, Microsoft, *Windows NT Server 4.0 End-user License Agreement*, *with* Microsoft, *Windows 2000 Server End User License Agreement* (on file with the author).

179. Apache Software Foundation, *Apache Software License*, *at* http://www.opensource.org/licenses/apachepl.php (last visited Dec. 20, 2002); *X11 License*, *supra* note 69. *But cf.* Perens, *The Open Source Definition*, *supra* note 2, at 183 (discussing the advantages of using the X license over both its progeny, the BSD and Apache licenses, and the GPL).

180. *See* Jim Hamerly et al., *Freeing the Source: The Story of Mozilla*, *in* OPENSOURCES, *supra* note

This proposed role is important because the goals and business objectives of open source developers are evolving rapidly.[181] OSLO could propose new license terms to deal with long-standing source licensing issues, such as the proper approach if someone asserts a patent against open source software.[182] OSLO could modularize[183] its standard form licenses and provide an analysis of each license provision to guide a developer's decision to deploy a given term in his or her license.[184]

In addition to creating and fixing licenses, OSLO could assist in ongoing questions about the interpretation of OSLO licenses. Today, interpreting the *de facto* standard open source licenses is one of the most confusing aspects of open source licensing. FSF often gives its view on what the GPL standard form license means, but its reviews are informal and usually rendered in private.[185] OSLO could improve on FSF's approach by setting up a formal process for considering interpretation questions, and OSLO could publish

---

2, at 197, 200-03 (discussing the development of the Netscape Mozilla open source license after it was determined the existing licenses were inadequate to meet Netscape's open source goals).

181. *See* Raymond, *The Magic Cauldron*, *supra* note 4.

182. *Compare* IBM, *IBM Public License Version 1.0*, *at* http://www.opensource.org/licenses/ibmpl.php (last visited Dec. 20, 2002) (granting a royalty-free patent license for any contributions to the Program regardless of the contributor), *with* FSF, *GNU-GPL*, *supra* note 6, at pmbl., § 7 (limiting distribution of the Program if the contributor is unable to grant a royalty-free patent license). *See generally* Bradley C. Wright, *Linux Users Risk Infringement*, INTELL. PROP. & TECH. L.J., Aug. 2001, at 1 (discussing the risks Linux users face); Dictmar Mueller & Graeme Wearden, *German judge puts SuSE Linux on hold*, ZDNET NEWS, Jan. 8, 2002, *at* http://zdnet.com.com/2100-1104-803681.html (last visited Dec. 20, 2002) (reporting a possible copyright infringement resulting from bundled Linux software); M. Craig Tyler & J. Wesley Jones, *Open-source Software Raises Licenses Issues*, NAT'L L.J., May 14, 2001, at C14 ("Simply put, if a proprietary software developer uses open-source code in a way that exceeds the scope of the open-source license agreement, he or she may be liable as a copyright infringer."). Although large companies such as I.B.M. hold many patents, nonprofit organizations obtain patents as well as and sue companies who infringe them. *See* Goldie Blumenstyk, *Cornell U. Sues Hewlett-Packard, Alleging Patent Infringement*, CHRON. OF HIGHER EDUC., Jan. 7, 2002, *at* http://www.chronicle.com/daily/2002/01/2002010707n.htm (last visited Jan. 7, 2002); *IBM receives the most 2001 patents*, ZDNET NEWS, Jan. 10, 2002, *available at* 2002 WL 4463554; *MIT Alleges Patent Violation*, SEATTLE TIMES, Jan. 5, 2002, at C4, *available at* 2002 WL 3886571.

183. Creating license templates such as this is common practice among licensing lawyers at technology companies. *See* Bezroukov, *BSD vs. GPL*, *supra* note 12, §§ 3.1-.2 (proposing a modular license that would change over the life of a program).

184. *See* Perens, *The Open Source Definition*, *supra* note 2, at 185 (providing barebones advice to other hackers on how to choose a license).

185. Hackers often believe that it is the FSF's responsibility to interpret what the GPL means as applied to their software. Certainly, the FSF knows what it meant when it created the GPL standard form license, and that is useful to know, but once a hacker applies the license to his or her software, it is the hacker's intention that controls. For example, Linus Torvalds apparently disagrees with FSF's expansive view of whether dynamic linking creates a "work based on a program." *See* Torvalds, *Introduction to Linux-KPL*, *supra* note 121 (rejecting the characterization of the Linux Kernel as a derived work).

formal advisory opinions setting forth its conclusions. It could also proactively educate the open source community about the best uses of OSLO's various license forms.

2. *OSLO's organization*

OSLO would be a place where programmers and lawyers come together to create useful licensing practices for open source software development. To be effective, it will need hacker participation as well as participation by the ever increasing number of commercial developers and users of open source software.[186] Input from developers of code that is not open source but needs to work with open source software is also important.[187] OSLO could act as a neutral turf where folks from the open source community and folks from the commercial software community would come together to discuss open source licensing.[188]

Until now, hackers have defined and often written open source licenses. They have done a credible job of drafting licenses, but it is not their craft. In the days when hackers wrote code primarily for other hackers, buggy licenses may have been little more than a nuisance. Now, open source software plays a significant role in the world economy,[189] and buggy licenses threaten to

186. I.B.M. and Hewlett-Packard, in particular, have embraced open source development. I.B.M. has pledged a billion dollars to development of open source code and H.P. hired hacker Bruce Perens to provide leadership its open source initiatives. Joe Wilcox, *IBM to spend 1 billion on Linux in 2001*, CNET NEWS.COM, Dec. 12, 2000, *at* http://news.com.com/2100-1001-249750.html (last visited Dec. 20, 2002); Stephen Shankland, *IBM: Linux investment nearly recouped*, CNET NEWS.COM, Jan. 29, 2002, *at* http://news.com.com/2100-1001-825723.html (last visited Dec. 20, 2002); Stephen Shankland, *HP hires Linux luminary*, CNET NEWS.COM, Dec. 4, 2000, *at* http://news.com.com/2100-1001-249387.html (last visited Dec. 20, 2002) (discussing hiring of Bruce Perens). *See also Week in review: In Linux land*, *supra* note 173 (reporting recent converts to Linux). Several large end users have embraced open source code. *See* Gomulkiewicz, *How Copyleft Uses License Rights*, *supra* note 7, at 185 (describing companies embracing open source software).

187. *See*, *e.g.*, MICROSOFT LICENSING, MICROSOFT CORP., *Questions About the GNU-GPL*, *supra* note 94 (discussing issues that can arise from integration of programs with open source GPL licensed programs). There may be fear among hackers that an open source rival, such as Microsoft, could attempt to commandeer a standards organization like OSLO. In some circumstances, such activity could run afoul of the law. See Lemly, *supra* note 153, at 1927-35. More importantly, however, no hackers would use OSLO's form licenses and, consequently, OSLO would be a failure.

188. *Cf.* Wong, *supra* note 157 (discussing how software companies came together to create uniform standards for XML, a software standard that allows the exchange of information over the internet).

189. *See* Matthew Broersma, *Raymond: Cheap PC's will doom Microsoft*, ZDNET NEWS, Feb. 28, 2002, *at* http://zdnet.com.com/2100-1104-847416.html (last visited Dec. 20, 2002) (discussing Eric Raymond's contention that open source software will drive the cost of personal computers below $350, jeopardizing Microsoft's continued viability).

impede the success of open source software. Hackers still have a valuable role to play in licensing, but lawyers need to step up their involvement.

OSLO could be organized in many ways. The W3C and IETF present potential organizational models that should be a comfortable fit with hacker culture.[190] A law school with expertise in licensing law could play a leadership role in OSLO much like the roles Massachusetts Institute of Technology, Institut National de Recherche en Informatique et Automatique, and Keio University play in W3C.[191] OSLO could draw on legal expertise from bar organizations such as the American Bar Association or the American Intellectual Property Law Association[192] and business expertise from organizations such as the Licensing Executives Society.[193]

VI. Conclusion

All software has bugs, as do all software licenses. Hackers pride themselves in rapidly fixing buggy software using the "many eyeballs" method described in the *Cathedral and the Bazaar*.[194] However, hackers have not created a method of fixing buggy licenses that is as successful as their method of fixing buggy software. An open source license standards organization would improve the buggy license forms that exist today and evolve open source licensing in the future.

190. *See supra* notes 119-25 and accompanying text. The formal processes of OSLO undoubtedly would be more streamlined than those of either IETF or W3C which are creating highly technical specifications.

191. W3C, *About W3C*, *supra* note 162 (listing the world-wide W3C leaders). Small, individual efforts have already begun. For example, Stanford law professor Lawrence Lessig has established the "Creative Commons" project to encourage sharing and use of copyrighted works ranging from software to music. John Borland, *Lessig plans digital rights organization*, CNET NEWS.COM, Feb. 11, 2002, *at* http://news.com.com/2100-1023-834775.html (last visited Dec. 20, 2002). See Creative Commons, About US, at http://creativecommons.org/learn/aboutus/ (last visited Dec. 20, 2002).

192. A.M. Intell. Prop. Law Ass'n, *Description of AIPLA*, *at* http://www.aipla.org/html/learn.html (last visited Dec. 20, 2002) (dedicated, in part, to "aid in the improvements in laws relating to intellectual property and their proper interpretation by the courts").

193. Licensing Executing Society (U.S.A. and Canada), Inc., *Mission Statement*, *at* http://www.usa-canada.les.org/aboutus/mission.asp (last visited Dec. 20, 2002) (functioning, in part "as a research organization and assist in furthering the employment of technology and the licensing and other transfer of technology").

194. Raymond, *The Cathedral and the Bazaar*, *supra* note 1.