The Myths And Realities Of Open Source Code Licensing: Business And Legal Considerations

BY HARRY RUBIN AND JASON ISAACS*

On its face, the many attractive features of open source licensing are readily apparent when juxtaposed against the traditional proprietary software model. For many technology companies, the adoption or rejection of the open source model may have potentially far-reaching and irreversible consequences. For both licensors and licensees, open source licensing presents numerous business and legal maelstroms requiring careful consideration and navigation. Following a discussion of the salient characteristics of open source licensing, we elucidate key factors which should figure prominently in a strategic approach to open source licensing.

I. Open Source and Free Source

Open source licensing arose out of the "free software" movement pioneered by MIT's Richard Stallman. "Free" refers to freedom of operation or use. It does not mean "free of charge." The free software movement advocates unrestricted distribution of software without the veil of secrecy imposed by the proprietary model. Stallman founded the Free Software Foundation in 1985 and developed the General Public License ("GPL") whose essence is protecting the licensee's rights to distribute, to modify and to use source code. Software companies typically rely on contractual, trade-secret, copyright and, increasingly, patent protections to safeguard the owner's proprietary interests in its software. The free software model uses "copyleft"—a term deliberately coined to underscore its contraposition to copyright—to protect the access to and use of the software for the open source community.

Freely distributing a program in the public domain allows anyone to share the program, but also allows licensees to incorporate the program into their own proprietary software. Therefore, once an unprotected program reemerges as part of a proprietary program, the recipients of the proprietary code no longer have the freedoms of use bestowed by the original author. Under the GPL's "copyleft" principles, any redistribution of software by the licensee, whether the software is modified or not, must be subject to the GPL. Accordingly, incorporation of copyleft code into a licensee's proprietary work requires the licensee to distribute its proprietary source code without proprietary protection.

Today's open source landscape reflects a schism between the "free software" movement and a more commercially friendly group that has adopted the open source moniker. Indeed, when Netscape announced its intention to give away its source code for its browser software in 1998, a group of software developers abandoned the "free software" movement altogether. Whereas the free software movement seeks to replace the proprietary model, the open source movement embraces the opportunities presented by the business world and seeks to entice commercial developers. Several open source licenses have emerged in recent years, including the Berkeley Software Distribution License ("BSD"), the Mozilla Public License ("Mozilla") and the Artistic License for Perl programming (the "Artistic License").

Elements of the Open Source movement eventually coalesced into the Open Source Initiative ("OSI"). OSI sought to align itself with the commercial software industry to spread the use of open source licensing and to develop ways to monetize its fruits. Bruce Perens, a prominent member of the open source Linux community and a consultant to Netscape, drafted a standard that eventually became the Open Source Definition ("OSD"). OSD sets forth nine requirements to which a license must conform to be deemed an Open Source license:

1. *Free Distribution*: The license may not restrict any party from giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee.

2. *Source Code*: The program must include source code and must allow distribution in source code as well as compiled form.

3. *Derived Works*: The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. *Integrity of the Author's Source Code*: The license may restrict source code from being distributed in modified form only if the license allows the distribution of 'patch files' with the source code for the purpose of modifying the program at build time.

5. *No Discrimination Against Licensee/Users*: The license must not discriminate against any person or group of persons.

6. *No Discrimination Against Fields of Endeavor*: The license must not

*Harry Rubin, Head, Global Technology Transactions Practice, and Jason Isaacs.

restrict anyone from using the program in a specific field of endeavor.

7. *Distribution of License*: The rights attached to the program must apply to all licensees without the need for execution of an additional license by those parties.

8. *License Must Not Be Specific to a Product*: The rights attached to the program must not depend on the program's inclusion in a particular software distribution.

9. *License Must Not Contaminate Other Software*: The license must not place restrictions on other software that is distributed along with the licensed software.

The GPL, BSD and Mozilla Public License all generally conform to the OSD. Conspicuously absent from the OSD is a copyleft requirement such as that found in the GPL. Indeed, at first glance, the OSD's non-commercialization edict (No. 9) seems contrary to the GPL's copyleft requirement which prevents a licensee from restricting other software distributed with the licensed software. The non-commercialization edict, however, is consistent with copyleft, as copyleft does not attach to other programs that are merely distributed or bundled with the same media. Copyleft only applies to software that contains or is derived from the open source licensed software.

II. Business and Legal Considerations

Licensors and licensees alike shall carefully weigh the competing salient legal and business considerations and trade-offs in deciding whether to adopt an open source model.

A. Business Considerations

(a) Access to Open Source Development:

Licensors justifiably are tempted by potentially harnessing the programming muscle of the open source community, leading to free aggressive development and shorter implementation timelines. For example, within a month after Netscape released its browser source code under an open source license, the open source community had completed a new version of the browser, including a function enabling secure Internet transactions.

(b) Revenues and Costs:

Several factors should frame the financial analysis of an open source strategy.

i. *Free Code*. Open source software is either licensed entirely free of charge or for a nominal amount. Services to support the open source code software usually cost much less than services for closed source software. Those dramatic savings are the primary attraction of open source licensing for licensees.

ii. *Transaction Costs*. Benefiting both licensors and licensees, the open source model dramatically reduces the transaction costs associated with licensing proprietary software. The open source license does not lend itself to negotiation. Its strength is precisely its uniform applicability to all users. While under the proprietary model end-use licensing for mass-market software is generally subject to standard and end-user licenses, license arrangements tend to be customized and heavily negotiated with enterprise customers and throughout distribution channels.

iii. *Business Models*: Notwithstanding the gratis licensing of open source software, software companies may generate substantial revenues by combining open source licensing with income generating models. These include: charging for related distribution, support or consulting services; providing the free open source software to help bolster license related proprietary software products which are licensed for profit; and profiting from hardware sold with the free software. Distributing software packages, which include both proprietary and open source products, is also a viable alternative. Here, the open source release is designed to create market familiarity and generate a robust development community. The commercially licensed and functionally superior closed source release will have more functionality and attract the open source licensees. Depending on a software company's particular business opportunities, the creative use of open source licenses actually could prove more profitable than the traditional alternative. Thus, *open source licensing does not condemn software companies to non-profit status*.

(c) Evolution and Innovation vs. Self-Perpetuation and Standardization:

A significant advantage for all parties is that open source software is constantly modified, debugged, enhanced and improved. The time lag between bug identification and delivery of bug fixes is dramatically reduced. Licensors may offer steadily evolving product lines to licensees, who would normally have to wait much longer for this type of development from closed source licensors. Not only is open source software better suited for rapid evolution, but it should ultimately afford licensees with access to a wide variety of derivative products.

A critically important and frequently overlooked characteristic of open source licensing is self-perpetuation. Licensees in the open source community enrich and enhance the code initially licensed by the licensor. The more useful the software, the more popular it becomes and the less likely it is that users will be inclined to migrate to different products. The firm entrenchment of certain software programs in the open source code community potentially converts certain programs to industry standards. (Lindows, RedHat and Sun are trying to create an industry standard). For the licensor, giving away software to create an industry standard may not bear fruit initially, but once market share has matured and a program becomes a standard, its longevity is ensured. Software companies whose business model calls for the frequent introduction of new products may find that their very success in creating a standard is a double edged sword. Dislodging their own previous product and generating demand for the new product

will become more difficult the more popular and successful the previous software product has become, especially if the new product renders the previous standard inferior or obsolete.

(d) Psychological and Technical Barriers to Adoption:

From a psychological perspective, open source development is a challenge of taboo breaking dimensions. Software development has traditionally been conducted in acute secrecy and subject to strict confidentiality, development, proprietary and nondisclosure agreements. Source code, always considered the "crown jewel," has been vigorously protected. Most source code disclosures occur only after a narrowly defined source code escrow release condition has been triggered. Open source licensing now requires software companies to eschew the very axioms which have governed software since the dawn of the industry and adopt the polar opposite approach. There also may be strong technical barriers to entry in adopting an open source model. Switching to a new operating system, or merely a new method of licensing, deters many from either participating in development or using such software. Of course, early adopting licensees will have the opportunity to adapt the software to their own needs from the beginning, and Licensors may market the opportunity to become an influential contributor to licensees wishing to participate in the development.

(e) Flexibility:

The proprietary model offers parties to a license agreement the freedom and flexibility to negotiate individually and to tailor the license terms to suit their respective needs. The open source model entirely lacks such flexibility. The open source model invariably utilizes a single licensing formulation to bind differently situated licensees, who may intend to use or to modify the software for different purposes. Such artificially imposed uniformity seldom provides licensees with a license meeting the licensees particular needs.

(f) Value:

The perceived value of software based on the open source model may be diminished, or eliminated, because the software's proprietary value is compromised.

(g) Compatibility:

Ensuring compatibility between proprietary software programs and platforms has always been a key customer objective. Access to source code enables the development of new software that will be compatible with a host of existing open source programs. Proprietary software usually cannot be developed in this fashion by separate entities unless each is given access to the other's source code. Preparing such license arrangements is costly, time consuming and only practical when a small number of entities is involved.

B. Legal Considerations

(a) Is It Legally Binding?

A threshold consideration is whether open source licenses constitute binding agreements. Most open source software is licensed under notice provisions without requiring a manifestation of contractual assent. Proprietary software is licensed under written agreements or, to mass market end-users, under "shrinkwrap" or "clickwrap" agreements requiring licensees to manifest assent to the license terms. Opening the shrinkwrap to use the software is deemed an act of assent to the encased license, because one cannot avoid seeing the license under the shrinkwrap. Similarly, a clickwrap license is an online agreement which requires the licensee to assent by clicking on an "I accept" button. Courts generally find such licenses enforceable where the licensee has the opportunity to express assent, or refusal to accept the license, and the license provides for an appropriate remedy, if applicable, such as refunding previously paid license fees. The Uniform Computer Information Transactions Act (UCITA) also validates such methods of assent as enforceable contracts.

Significantly, the GPL and BSD only require that a notice regarding the license accompany the software. Notice alone may not legally suffice to satisfy the assent requirement. *Courts may, therefore, find that open source licenses that merely rely upon notice provisions to bind licensees do not create a legally enforceable agreement. Therefore, open source licensors should make the assent requirement explicit by at least utilizing a clickwrap, shrinkwrap or similar mechanism*.

(b) Enforcement of Intellectual Property Rights:

The advantages of employing programmers far and wide to improve one's program may be offset by the difficulty of trying to organize those same programmers to defend one's intellectual property rights. It may be hard to establish what the "official" version of the software is and who has a right to enforce the copyrights underlying the software. There is ample potential for dispute when the "owner" is actually a group of numerous and widely dispersed programmers. Indeed, in an unusual move for the open source community, the Free Software Foundation now requires that authors assign their rights in their software to the Free Software Foundation to allow a single entity to enforce the licenses and act as a watchdog.

(c) "Work Made for Hire" Doctrine:

Under the "work made for hire" doctrine, if code is developed by a programmer during the course of employment, the employer owns the code. Moreover, the programmer likely will be subject to a proprietary information and inventions agreement that would prevent the programmer from complying with the terms of an open source license. Under the proprietary model, it is far easier to conduct due diligence to ascertain who owns the program.

(d) "The Trojan Horse"—
Viral Software:

Free and open source licenses create the *illusion of freedom for the licensee*. Incorporation of copyleft code into a proprietary work requires the licensee to distribute freely its proprietary work's own source code to the public. The GPL

states: "You 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." Thus, the "viral" open source code "infects" the proprietary code.

Consequently, many software companies do not allow licensees to incorporate open source software into their proprietary software or even use open source software in conjunction with their proprietary software. Likewise and conversely, many licensees insist on a warranty that the licensor's software does not include open source code, because the licensees are afraid to infect their own software with the open source code.

A copyleft provision can be a short single sentence. Its subtle phrasing and seemingly innocuous presence may be overlooked, leading to the loss of the intellectual property rights in proprietary software. Variants which closely mirror one of the non-copyleft licenses, but insert a copyleft clause, require equal vigilance.

(e) Software Patents:

Licensors increasingly seek patent protection to bolster their proprietary hold over their software. Significant patent claims could severely restrict the usefulness of open source software to the licensee. Open source development programs may fall under patent claims and be deemed infringing, forcing open source licensees to obtain a patent license. The GPL tries to solve the problem by having licensees agree not to apply for patents for their developments under the GPL. Such a restriction may be difficult to enforce.

(f) Legal Inadequacy:

While many of the most popular open source licenses were drafted with legal guidance, they do not cover pivotal terms typically found in standard licenses. The GPL (and most open source licenses) does not specify the duration of the license granted and omits key provisions, such as indemnification, warranty, assignment, choice of law and dispute resolution.

(g) Uncertainty:

The law regarding open source licenses remains sparse and unclear, offering little or no guidance to potential open source developers or users. The global nature of the software industry and the "virtual location" of the open source community substantially obfuscate matters further, raising complex issues of conflicts of law, dispute resolution and enforcement.

Conclusion

We deliberately avoided proffering any prescriptive judgment about open source licensing. Indeed, formulaic rules for the adoption or rejection of the open source model are deleterious. As business objectives and exigencies vary widely among software companies and users, open source licensing should not hinge on generalizations or oversimplified predispositions. Rather, each licensee and licensor