

# Practical Strategies For Developing Open Source Compliance Programs

---

Jason Haislmaier



## Jason Haislmaier

is a partner in the Intellectual Property Group of the law firm of Holme Roberts & Owen, in Boulder, Colorado. He is also a Professor of Copyright Law at the University of Colorado Law School in Boulder. He concentrates his practice on representing emerging and established technology companies in technology and intellectual property issues, with a special area of emphasis on helping clients develop and implement open source software compliance strategies and contend with other open source licensing issues. This article is based on a paper the author prepared for a seminar sponsored by the ABA's Section of Intellectual Property. He can be reached at [jason.haislmaier@hro.com](mailto:jason.haislmaier@hro.com) and followed at <http://twitter.com/haislmaier>.

---

**With the right program, you really can close the open source compliance gap.**

---

**“WHY SHOULD I** be concerned about compliance with open source software licenses?” This question was once asked predominantly by companies in the computer software and other technology industries. With the unprecedented growth in the usage of open source software this question is now, however, relevant to all companies that use or develop software, regardless of their industry. Most of these companies have knowledge of the proprietary software they license from third parties and understand the importance of complying with the terms of the proprietary software licenses applicable to this software. They have implemented procurement procedures requiring identification of proprietary software licensed from third parties and review of applicable proprietary software licenses. They have also established internal controls to help ensure compliance with these terms throughout the term of the applicable license. However, it seems that far fewer companies are aware of whether they are using open

source software and understand the importance of complying with applicable open source licenses (and the consequences of non-compliance). Unlike with proprietary software, they have not implemented procedures to identify open source software entering their organizations or controls intended to ensure open source license compliance. This open source software “compliance gap” places companies at an increased risk of unintended violations of open source licenses and other potential liabilities. Given the potential severity of these consequences, any company that uses or develops software should understand the risks posed by the open source software compliance gap and take steps to close that gap. This article identifies practical strategies that companies can use to identify and close the open source compliance gap in their organizations.

## THE LEGAL NATURE OF OPEN SOURCE

• Open source software can be described on many levels. On one level it is a pow-

erful organizational tool for collaborative software development and maintenance. On another it is recognized as an ideological set of beliefs regarding the availability of source code and the commercialization of computer software. From a legal perspective, open source software refers to software licensed under an “open source” license. Open source software is distinguishable from proprietary software as well as from freeware and public domain software. Unlike freeware and public domain software, open source is subject to a license agreement that places legal obligations on the licensee of the software. As a result, open source software should never be mistaken as being “free” from legal obligations. But the legal obligations imposed by open source licenses are significantly different from those imposed by traditional proprietary software

licenses. Traditional proprietary software licenses generally place significant restrictions on the use of software by the end user, in particular the source code to the software. In contrast, open source licenses create a more “open” legal environment generally characterized by the availability of both source and binary code versions of the software, the right to modify the software and the right to distribute those modifications.

## The Open Source Definition

The boundaries of the “open” legal environment created by open source licenses are defined by the Open Source Definition. *See*, Open Source Initiative, Definition, available at <http://www.opensource.org/docs/osd>. The Open Source Definition is promulgated by an organization called the Open

Source Initiative (OSI). *See also*, Open Source Initiative, available at <http://www.opensource.org>. The OSI approves licenses as “OSI Certified” based on

their compliance with the Open Source Definition. Once OSI certified, licenses are generally recognized as being “open source” licenses. The Open Source Definition establishes a number of criteria that a license must meet before it is considered an open source license. *Id.* Additional criteria of the Open Source Definition require:

- Source code integrity (the license may require that derived works of the software be distinguished from the original program);
- Nondiscrimination (the license must not discriminate against any person or group, or against use in any specific field of endeavor);
- Distribution of license (the license must apply to all to whom the program is distributed without the need for execution of an additional license);

---

**Unlike freeware and public domain software, open source is subject to a license agreement that places legal obligations on the licensee of the software. As a result, open source software should never be mistaken as being “free” from legal obligations.**

---

- Non-product specific (the rights attached to the program must not depend on the program's being part of a particular product);
- Non-restrictive (the license must not place restrictions on other software that is distributed along with the licensed program); and
- Technology neutral (the license may not be predicated on any individual technology or style of interface).

The core criteria of the Open Source Definition include the following:

- Source code availability. The license must allow for distribution of the software in source code (human readable) and object code (machine readable) forms. If the software is distributed only in object code form, there must be a "well-publicized" means of obtaining the source code and object code for no more than a reasonable reproduction cost. The source code must be in the preferred form for modification by a programmer;
- Free redistribution. The license will not prohibit anyone from licensing the software as part of an aggregated offering containing code from other sources, nor require that a royalty or other fee be paid in exchange for the licensed software;
- Derived works. The license must allow modifications and derived works of the licensed software and must allow any modifications or derived works to be distributed under the same license terms as the original software.

The Open Source Definition functions in much the same way as a technical specification by establishing a set of criteria that allow for multiple implementations that all meet (and are "compatible" with) the specification. The breadth of the criteria described by the Open Source Definition has allowed for a wide variety of licenses to be classified as open source. In fact, at the time this article was written there were nearly 70 different open source licenses approved by the OSI as meeting the Open

Source Definition. *See*, Open Source Initiative, Licenses (available at <http://www.opensource.org/licenses/>).

Some of the more well-known of these licenses include the General Public License (GPL), Lesser General Public License (LGPL), MIT, BSD, and APACHE licenses. In addition to the OSI-approved licenses, countless other licenses exist that identify themselves as being open source licenses but that have not been actually approved by the OSI. Some of these unapproved open source licenses are based on one or more of the OSI-approved licenses and appear to contain terms that comply with the Open Source Definition. Others, however, contain terms that diverge significantly from the requirements of the Open Source Definition.

### **Unapproved Open Source Licenses**

Given the variety of OSI-approved open source licenses (not to mention the many additional unapproved open source licenses), the potential implications of using, modifying, and distributing open source software varies greatly from license to license. While a detailed examination of the variety of open source licenses is beyond the scope of this article, an illustrative example is the distinction between copyleft and non-copyleft open source licenses. "Copyleft is a general method for making a program free software and requiring all modified and extended versions of the program to be free software [...] Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it." *See*, <http://www.gnu.org/licenses/licenses.html#WhatIsCopyleft>. Much has been made of the so-called "viral" effect of copyleft licenses such as the GPL or LGPL. These licenses contain terms that purport to obligate a licensee to make publicly available the source code of proprietary software that is distributed with open source licensed on these licenses. Free Software Foundation, Inc., GNU General Public License, Version

2 (June 1991), available at <http://www.opensource.org/licenses/gpl-license.php>; GNU Lesser General Public License, Version 2.1 (February 1999), available at <http://www.opensource.org/licenses/lgpl-license.php>. Non-copyleft licenses, such as the BSD and MIT licenses do not contain viral provisions and generally only obligate the licensee to provide certain attributions and notices when redistributing the software. *See* Open Source Initiative, MIT License, available at <http://opensource.org/licenses/mit-license.php>, and New BSD License, available at <http://www.opensource.org/licenses/bsd-license.php>.

Given the potentially serious implications of overlooking these differences, one should never assume that the terms of open source licenses are created equal and should instead understand the terms of any particular open source license before using software obtained under that license.

**UNIQUE RISKS OF OPEN SOURCE** • Companies that use third-party software are finding it increasingly unrealistic to avoid the use of open source software entirely. Even if a company has opted to ban the use of open source, they often find that open source software is being used within the company, or even in the company's products, in some form. For example:

- Developers may introduce open source code into their development projects without the knowledge of management;
- Third-party software licensed by the company may include open source software; or
- Domestic or offshore software development outsourcers may incorporate open source code into projects for the company.

As a result of the growing ubiquity of open source software, it is imperative that companies under-

stand the risks that arise when open source software is used within a company or incorporated into a company's products and take steps to mitigate these risks. Several of these risks include:

- Questions of interpretation. Not only do the terms of open source licenses differ greatly in their effect, but the terms of many open source licenses are notoriously unclear and ambiguous. Despite the fact that accepted interpretations have begun to develop, there remains a good possibility that a licensee and licensor to an open source license will each arrive at a different interpretation of the terms of the license and that a dispute over those terms will ensue;
- Uncertain rights due to undocumented chain of title. The model for open source software

---

**Companies that use third-party software are finding it increasingly unrealistic to avoid the use of open source software entirely.**

---

development is a community process through which developers make individual contributions to a collective software development project. (The open source "community process" refers broadly to the collaborative process through which loosely connected individuals come together largely via the Internet and make contributions to the development of various open source projects under a common open source license, and the online venues, such as <http://sourceforge.net>, on which the projects are hosted and maintained. *See, e.g.,* [http://en.wikipedia.org/wiki/Open\\_source\\_movement](http://en.wikipedia.org/wiki/Open_source_movement).) Many times, there is not sufficient legal review of these individual contributions to determine if they infringe on the intellectual property rights of third parties. The rights that a licensee obtains to the open source software developed through this process are only as strong as the rights of the contributors to their individual contributions. If any contributor does not have proper rights to their contribution, by implication, any licensee will also not

have proper rights in the open source software containing that contribution. As the number of contributors to any open source software project increases, so do the chances that one of the contributors did not have sufficient rights to make their contribution;

- Risks posed by the lack of representation and warranties. Most proprietary software licenses provide some form of representation and warranty addressing issues such as the operability of the software and intellectual property infringement. Many also provide indemnifications against many of these same issues. Most open source licenses, however, do not provide representations and warranties or indemnifications. The lack of representations and warranties and indemnifications leaves open source licensees potentially exposed if any of the issues commonly addressed in proprietary software licenses arises under an open source license.

### **WHY THESE RISKS SHOULD MATTER TO YOU**

• These risks are not just abstractions or distant contingencies. Things are changing at a very rapid pace with respect to open source. The law regarding enforcement of the licenses is developing on a daily basis. The penalties for breach can be severe and injunctions are not impossible or even unlikely. These topics deserve in-depth attention, so they will be the subject of another article scheduled to appear in the December issue.

### **PRACTICAL STEPS TO BUILDING AN OPEN SOURCE COMPLIANCE PROGRAM**

• What should you do to comply? The risks posed by using open source software are often complex. Implementing an effective open source compliance program allows companies to gain vision into their use of open source software and to reduce their exposure to these risks. But, there is no one-size-fits-all approach to open source compliance programs. Compliance programs that are too formulaic will

often fail to provide adequate focus on the particular risks facing the company. Compliance programs that are too detailed often prove impractical to implement with available resources and never reach completion. Striking the right balance requires that companies understand the nature and complexity of their particular open source software usage, determine the level of compliance appropriate for the company, and set the scope of their compliance programs accordingly.

While striking this balance is not an easy task, best practices shared by successful open source compliance programs are emerging. Experience shows that it is essential that companies tailor their open source compliance programs to leverage these emerging best practices. The following section of this article provides information on developing and implementing an open source compliance program based on these best practices.

### **Develop An Open Source Risk Profile**

Each company has its own tolerance level for the risks posed by open source software. This tolerance can vary greatly from company to company based on a company's business model and the potential effect the risk may have on that model. *See*, Stephen Mutkoski, *Seven Steps to Addressing Open Source Issues in Software Development*, (Practicing Law Institute Publications, available at: [http://www.pli.edu/emktg/toolbox/Seven\\_Steps1.pdf](http://www.pli.edu/emktg/toolbox/Seven_Steps1.pdf)). Before making business decisions regarding the use of open source, companies should develop an open source risk profile by evaluating the risks posed by open source to the company and the company's tolerance for these risks. The evaluation should consider unique risks posed to the company based on its business model, as well as the general risks posed to all users of open source. It should also consider key areas of the company's business model, such as core software products or areas of existing or future revenue for the company. For example, does the company generate a large percentage of its revenue either

directly or indirectly from software licensing? Does it instead rely on revenue from hardware or related services and maintenance? Does the company maintain a large portfolio of patents? Are those patents related to software? What revenue, if any, is obtained from licensing those patents? In developing a profile, companies should take care to ensure that the factors chosen accurately reflect the company's business model, and its overall tolerance for risk. Companies may also want to consider weighing each factor to ensure that the profile is focused on high-risk areas and does not place unnecessary emphasis on lower-risk areas.

Companies should use their open source risk profile to help evaluate the risks of using open source in light of the benefits and competitive advantages. Based on this evaluation, many companies will find it acceptable to use open source in some form. Of these companies, most will also find that their level of comfort for using open source varies based on the nature of the use. For example, a company may only be comfortable using open source licensed under certain open source licenses. Even then, that company might only be willing to accept the risks posed by a given open source license under certain limited circumstances. Based on this evaluation, the company may set certain rules for the use of open source in particularly sensitive "core" projects or areas of their business model in which open source risk tolerance is lower and apply different rules for other areas. The company might also elect to initially target its open source compliance program in areas of particularly great risk (and particularly low risk tolerance) and only later give more even attention across all areas of its business. In other situations, the company may find that the risks of using open source software come at too high a cost and are not acceptable under the company's risk profile. As a result, the company might make a reasonable business decision to pursue alternatives such as licensing or developing a proprietary solution to be used in lieu of the available open source option.

### **Conduct Open Source Audits**

Mitigating the risks posed by open source requires gaining knowledge of when and how open source software is (or may be) used. Open source audits are key tools in gaining this visibility. Before implementing an open source compliance program, open source audits allow companies to develop a baseline picture of their open source software usage. Later, open source audits in connection with the release of new products or in advance of a business transaction can allow companies to corroborate previously documented uses of open source. Audits at other times (even randomly) can help to ensure that applicable open source compliance policies are being followed. In any case, audits should be scheduled well before deadlines for the release of products or the completion of transactions to avoid last-minute delays if gaps in compliance are found.

The audit process should be tailored to the company, its business model, and the scope and nature of the company's open source usage. However the audit is structured, the goal of the audit process should remain focused on inventorying the nature and context of all open source usage within the company. This will generally include, at minimum, the name of the open source software itself, the relevant open source licenses, the source from which the software was obtained, and the manner in which the software is being utilized by the company. For example, is the open source software being used internally or distributed? Is it used "as is" or is it being modified? Does it link to or otherwise interact with other proprietary software of the company or its licensors? If so, what is the nature of the link or other interaction? Other information may also be relevant and companies should not view this or any other published list as being exhaustive. Many companies find it efficient to collect this information through questionnaires, surveys, and interviews of IT and development personnel. Information should also be solicited from third-party vendors of soft-

ware products licensed by the company to check for open source used in those products. Automated tools are now available that will review the actual software code used by companies and scan the code for instances of open source. These products generally operate by comparing the scanned code against a database of known open source code. While these automated tools can be a useful part of any open source audit, they are not yet sufficient to be relied upon as the sole basis for the audit and should be used only as part of an overall open source compliance program.

### **Create An Open Source Policy**

A consistent and coherent open source compliance policy is at the core of every successful open source compliance program. The policy should address the common open source scenarios that a company is likely to face. These scenarios can include requests from developers to use open source, requirements for maintaining records regarding open source review and usage, procurement of third-party software containing open source, and treatment of open source in corporate transactions. A compliance policy may also address other scenarios such as participation by the company in the open source community through release by the company of software under an open source license, employee contributions to open source projects, or company sponsorship of open source projects. Depending on the company and its business model, other company-specific scenarios may also prove relevant and each policy should maintain the flexibility to address all scenarios relevant to the company.

### ***Open Source Review And Approval Process***

Companies that decide to allow the use of open source should establish a process for reviewing and approving requests from within the company to use open source. The primary goal of this process is to make each use of open source an intended and

compliant use of open source. The review process begins with a request by a developer to use open source. It is best if the review process requires that requests be submitted well in advance of planned implementation or release dates. This will allow for the analysis to take place in real time and avoid the need for an after-the-fact “hindsight” analysis. It will also minimize potential delays in the software development process and potentially in other business transactions if the use is not approved and alternatives must be located.

Requests are typically submitted on standardized forms. The form should include basic questions regarding the proposed use, as well as the developer and project name, name and version of the software, applicable license, date of first use, etc. Additional questions should be drafted to address issues of particular importance given the company’s business model and risk profile. While these questions will vary from company to company, relevant issues include:

- The nature of the open source being used (e.g., widely used and well established vs. relatively unknown and untested);
- The business area in which open source will be used (e.g., regulated vs. unregulated);
- The product or service in which the open source will be used and how that product or service will be used by the company (e.g., key commercial product vs. prototype; distributed vs. internal); and
- The manner in which the open source will be implemented (e.g., modified vs. unmodified, standalone, statically linked, dynamically linked, etc.).

The compliance policy should provide details regarding the company’s position on each of these questions. This information will provide a tool for guiding the evaluation of each request. Companies will often find that they do not have the professional expertise and judgment in-house to adequately

develop an open source compliance policy. Many companies enlist outside counsel or consultants experienced with the issues involved with open source software compliance to assist in the process. As experience is gained within the company, the compliance policy itself should be revisited periodically to determine whether amendments and updates should be made to reflect current thinking within the company. For example, many policies will require an individual review of each request to use open source. Over time, however, a company may develop enough comfort to amend the policy to provide pre-approval to certain designated open source usage scenarios.

The compliance policy should also specify how the review will be conducted and who will be responsible for the review. Policies vary in this regard from requiring review by a centralized open source compliance team to distributing the reviewing authority to separate groups or individuals throughout the company.

### **Documentation Of Open Source Usage**

The results of any request to use open source should be carefully documented. If the open source is used, the steps required to comply with the applicable open source license should also be documented. Consistent documentation helps to track overall open source usage and can prove essential when it becomes necessary to later review previous compliance efforts. For example, compliance information will be required before the release of a product using open source for the purpose of including applicable notices, attributions, and disclaimers in documentation and taking any other steps necessary to comply with the applicable open source licenses. Likewise, information relating to particular open source licenses is increasingly requested by potential licensees before they will enter into a license to software containing open source. Information on company open source usage is also commonly required to be disclosed during mergers and acquisitions and

other significant corporate transactions. The process of having to recreate this information can be time-consuming and costly. Many companies find it convenient to maintain all records relating to open source review and usage in a single location, often in the form of a centralized database, regardless of whether the review process itself is centralized. Such a system can make it easier to comply with requests for information and to audit compliance with applicable open source policies.

### ***Software Procurement And Licensing Procedures***

Given the unique nature of open source, most if not all, existing software procurement procedures leave gaps when it comes to open source software. Worse, existing procedures may not be sufficient to deal with open source software at all. Accordingly, existing procedures should be updated, and new procedures implemented, to address the acquisition and use of open source. For example, procedures for procuring software should be updated to require that licensors disclose information regarding all open source software that is part of the software being licensed. Agreements relating to the purchase, license, and outsourced development of software should also be updated to reflect these procedures and to ensure that other provisions of these agreements (such as representations and warranties) are broad enough to cover the risks posed by open source software. The information obtained from licensors should be at least as extensive as the information required under the company compliance policy had the use originated as a request from within the company and should, when possible, be obtained without a requirement of non-disclosure.

Licenses of software granted by the company also require that the impact of open source software be taken into account. For example, most open source licenses prevent anyone from making the open source subject to the terms of proprietary agreement. As a licensor of software that makes use

of open source, the company should include open source-specific license provisions in any agreement under which such software is licensed. These provisions should ensure that any open source included in the transaction is not inadvertently made subject to the terms of the agreement. The provisions should also clarify that any open source is subject to the specific open source license accompanying that software and not subject to the proprietary agreement being entered into between the parties. The company may also consider whether it will expressly exclude open source from the general representations and warranties and indemnifications covering the licensed software.

### ***Corporate Transactions Practices***

Company policies regarding merger and acquisition transactions may also need to be updated to account for open source, particularly if the company's business model includes growth through acquisition. When the company is acting as the acquirer, representations, warranties, and other provisions in the purchase agreement should require the disclosure of all open source software that is subject to the transaction. Diligence practices should also be updated to require open source disclosure and include guidance regarding the review of any disclosed open source software and licenses. If the company is considering a merger or acquisition as an exit strategy, it should structure its compliance policy so that it will be able to offer a level of disclosure and provide representations and warranties regarding its own open source usage that will be acceptable to a potential acquirer.

### ***Participation With The Open Source Community***

Companies that make more extensive use of open source may want to address in their compliance policies interaction and participation between the company and the open source community. This includes release by the company of software under

an open source license, employee contribution to open source projects, or company sponsorship of open source projects.

It is increasingly the case that developers may be involved as contributors to some form of open source development project. Most companies require that employees sign proprietary information and invention assignment agreements stipulating that the company owns any work product produced by the employees in the course of their employment. If an employee who is subject to such an agreement makes a contribution to an open source project, there is a real possibility that the contribution may in fact include company code or implicate company trade secrets, patents, or other intellectual property rights. While many companies have realized benefits from allowing employees to contribute to open source projects, the loss of valuable company intellectual property is not one of them. Many employees may not appreciate the potential implications and risks posed by this scenario. Companies should consider whether they are at risk and establish a process for reviewing and approving developer participation before it takes place.

Companies are also increasingly electing to become directly involved with the open source community. (One way in which companies do this is through the release by the company of software under an open source license and sponsorship by the company of open source development projects.) The decision by a company to directly engage the open source community in one of these scenarios involves many considerations. What license will the software be released (or developed) under? Where will the development be hosted? What additional resources will be devoted to the success of the release/development? What steps are being taken to ensure the integrity of the company's other intellectual property rights? The way in which these questions are answered can be vitally important to the success of the particular project. For companies making these decisions, compliance policies should

establish guidelines that must be followed before the company can move forward with the project.

### ***Field An Open Source Compliance Team***

Successful implementation of an open source compliance policy requires the combined efforts of employees from all groups involved with open source within the company. Accordingly, many companies field an open source compliance team to aide in managing the company's open source compliance efforts. Compliance teams are generally cross-functional and are staffed by representatives from management, software development, and legal. Teams can also include other relevant groups such as risk management and procurement. Companies may also choose to include outside counsel or consultants experienced with the issues involved with open source software compliance as adjunct members of the team. The team typically serves as the central point of contact for open source activities within the company, including:

- Administering and overseeing the company's open source compliance policies;
- Directing and coordinating open source compliance efforts between the various business and project teams within the company;
- Providing information and expertise regarding open source issues faced by the company; and
- Managing and updating the open source compliance program and open source review process.

The role of the team should be flexible and evolve as the company's use of open source grows. For example, as the company begins to engage the open source community, the team can also serve as the external voice of the company in open source licensing activities and communications with the community. Regardless of the mission of the team, it should have a strong and open mandate from executive management to fulfill that mission.

### **Develop Open Source Education And Training**

Ongoing education and training is an essential part of implementing and gaining acceptance for an open source compliance program. Overall, education and training programs should foster and promote a "culture of compliance" for open source throughout the company. Training programs should begin by building a common awareness and understanding among employees regarding the issues relevant to open source. They should emphasize the facts regarding open source while working to dispel some of the many myths and misconceptions. The programs should also educate employees regarding the policies and procedures of the compliance program, while working to publicize and promote adoption of the program itself.

Training and education should include all stakeholders involved with open source compliance within the company. Care should be taken to tailor the programs to fit the particular backgrounds and experience of each of these groups. For example, lawyers and procurement personnel will likely require training focused on the architecture and design of the company's software. Software developers and architects, however, will likely require training regarding the legal nature of open source software and the business risks posed by using open source software. Care should also be taken to foster an atmosphere of inclusion among all stake holders. While there may be a tendency for certain groups, such as software developers, to see open source compliance as yet another example of management and legal intruding on their territory and impeding their ability to do their jobs, careful implementation can help to lessen this reaction and foster a more collaborative atmosphere.

The training itself can include formal and informal methods. More formal instructional sessions can involve lectures by in-house and outside legal counsel and other open source professionals regarding the various aspects of open source software

and the risks posed by open source to the company. These sessions can also include discussions of the company's open source compliance program and its impact on existing and planned company uses of open source. Less formal training may include smaller group question and answer sessions where information is exchanged and ideas are developed between interested individuals. The ideas generated through these sessions should be recorded and, when applicable, used to update training materials and improve the open source compliance program itself.

### **Implement Enforcement Mechanisms**

As with any compliance program, an open source compliance program is of little value unless it is enforced. Open source compliance programs should include mechanisms for ongoing monitoring of adherence to the program and for enforcing open source policies and procedures throughout the company. This includes enforcement in the case of ongoing software development projects, but should also include review of completed projects to ensure that circumstances have not arisen that would alter the outcome of the review. For example, compliance obligations for software licensed under the GPL that when reviewed was only used internally will change dramatically if that software is later distributed to customers, particularly if the software is being distributed in a modified form or as a derivative work of other software. These efforts should also facilitate a consistent policy implementation, for example, by ensuring that new employees are adequately trained in the company compliance program and that key roles in the compliance program remain filled regardless of employee turnover.

### **Periodically Update Existing Programs**

The scope and nature of a company's use of open source is not likely to remain static. In companies that use open source, existing uses of open source often give rise to additional uses, and new companies or products may be acquired and bring with them new uses of open source. Even in companies that choose to prohibit the use of open source, exceptions may be made to rules prohibiting the use of open source or uses may occur without being recognized. At the same time, the open source community itself also continues to move forward as interpretations regarding open source licenses and existing legal risks evolve. In this environment, an open source compliance program must also evolve to address this changing landscape. Companies should address this by monitoring open source activities within the company and in the open source community and building the opportunity for periodic reviews and updates into their compliance programs.

---

**As with any compliance program, an open source compliance program is of little value unless it is enforced.**

---

**CONCLUSION** • The continued acceleration in the use of open source software has made developing and implementing an open source compliance program a business necessity. While no one form of compliance program will be effective for all companies, the emerging best practices discussed in this article have proven successful for a number of companies. Whether companies handle the task of developing a compliance program alone or in partnership with legal and other professionals experienced with open source compliance, the key is to begin a process that yields a compliance program tailored to fit the size and needs of the company. Companies that elect not to follow this path will face the prospect of an increasingly signifi-

cant open source compliance gap as open source usage and the expectations regarding open source compliance continue to rise. Companies that develop and implement an open source compliance plan are, however, far more likely to close the open source compliance gap. By closing the gap they are also likely to realize benefits far beyond just mitigating the risks posed by the open source. They

will position themselves to capture and leverage the value of open source in their organizations in terms of decreased costs, increased market share, and greater agility to meet customer demands. Today more than ever before, implementing an open source compliance policy is instrumental in helping companies close the open source compliance gap and realize these benefits.

**To purchase the online version of this article—or any other article in this publication—go to [www.ali-aba.org](http://www.ali-aba.org) and click on “Publications.”**

### **PRACTICE CHECKLIST FOR Practical Strategies For Developing Open Source Compliance Programs**

- Develop an open source risk profile. The evaluation should consider unique risks posed to the company based on its business model, as well as the general risks posed to all users of open source. It should also consider key areas of the company’s business model, such as core software products or areas of existing or future revenue for the company.
- Conduct open source audits. Before implementing an open source compliance program, open source audits allow companies to develop a baseline picture of their open source software usage. Later, open source audits in connection with the release of new products or in advance of a business transaction can allow companies to corroborate previously documented uses of open source.
- Create an open source policy. The policy should address the common open source scenarios that a company is likely to face.
- 
- Establish an open source review and approval process. The primary goal of this process is to make each use of open source an intended and compliant use of open source. It is best if the review process requires that requests be submitted well in advance of planned implementation or release dates. While these questions will vary from company to company, relevant issues include:
  - \_\_\_ The nature of the open source being used (e.g., widely used and well established vs. relatively unknown and untested);
  - \_\_\_ The business area in which open source will be used (e.g., regulated vs. unregulated);
  - \_\_\_ The product or service in which the open source will be used and how that product or service will be used by the company (e.g., key commercial product vs. prototype, distributed vs. internal); and

---

\_\_\_ The manner in which the open source will be implemented (e.g., modified vs. unmodified, standalone, statically linked, dynamically linked, etc.).

- Document open source usage. The results of any request to use open source should be carefully documented. If the open source is used, the steps required to comply with the applicable open source license should also be documented. Be sure to document:
  - \_\_\_ Software procurement and licensing procedures;
  - \_\_\_ Corporate transactions practices;
  - \_\_\_ Participation with the open source community.
- Implement enforcement mechanisms. This includes enforcement in the case of ongoing software development projects, but should also include review of completed projects to ensure that circumstances have not arisen that would alter the outcome of the review.
- Periodically update existing programs. The scope and nature of a company's use of open source is not likely to remain static. In companies that use open source, existing uses of open source often give rise to additional uses, and new companies or products may be acquired and bring with them new uses of open source.