GETTING WITH THE PROGRAM:
A GUIDE FOR LAWYERS WORKING WITH FREE AND OPEN SOURCE SOFTWARE IN THE ENTERPRISE

Gwyn Firth Murray
Matau Legal Group
www.mataulegal.com
October 2005

GETTING WITH THE PROGRAM:
A GUIDE FOR LAWYERS WORKING WITH FREE AND OPEN SOURCE SOFTWARE IN THE ENTERPRISE

Gwyn Firth Murray
Matau Legal Group © 2005

INTRODUCTION

As recently as five years ago, there were very few lawyers around who knew – or cared – anything about free or open source software (together, "FOSS"). The prevailing view among corporate attorneys seemed to be that FOSS was "a passing phase" among engineers and that we software lawyers needed to practice "real" law -- i.e., law relating to the licensing in and out of proprietary solutions. Most proprietary technology companies seemed to have their heads in the sand, hoping that FOSS would just go away and leave them alone.

When I joined VA Linux Systems as its first lawyer in 1999, it became instantly and abundantly clear to me that lawyers – and commercial enterprises -- were at risk of missing an important boat here. FOSS development and the FOSS developer community (the "FOSS Community") were going, and growing, strong. Engineers were writing, modifying and distributing software on terms they either wrote or interpreted themselves. And we lawyers were to be left behind if we didn't start learning about, and adapting to, the reality of FOSS.

Times have changed, but many proprietary high tech companies – and the lawyers that work for them – haven't quite caught up yet. We are getting there, though. As FOSS has infiltrated the corporate enterprise, more and more lawyers are recognizing the importance of understanding FOSS and the legal issues surrounding its use. Companies and lawyers are realizing that, whether they like it or not, FOSS is a part of their world and they need to both understand and manage it.

± Gwyn Firth Murray was one of the "early adopters" in the legal community of a focus on open source software. She is founder and principal of the Matau Legal Group, which offers a broad range of commercial, licensing, and other legal services to both start-up and established companies in the high tech and biotech industries (see www.mataulegal.com). She also is co-founder of Open Bar, Inc., a not-for-profit organization focused on legal rights and responsibilities in the world of open source software (see www.open-bar.org). Ms. Murray is a graduate of Stanford University Law School, and also holds an M.A. in Latin American Studies from Stanford University. She obtained her B.A. magna cum laude and with distinction in economics from Yale College.

FOSS BEST PRACTICES IN AN IDEAL WORLD

Only a year ago, there was limited information available about how to manage FOSS in the enterprise – either from a business or a legal perspective. Lots of "Fear, Uncertainty and Doubt", or "FUD", was gravitating around about how to manage the inflow of FOSS into proprietary technology companies. One of the most frequent demands I heard in 2004 for FOSS seminar topics was "Best Practices" relating to licensing in of FOSS, particularly in the context of mergers and acquisitions. Within the past year or so, that call for information has been answered, with much discussion and writing about "best practices" for FOSS.[1]

In general, the consensus is that, in an ideal world, every high tech company would have in place a culture, policy and process for managing the influx, use and outflow of software. Elements of such an ideal program are:

- The company's executive team and all relevant company functions (i.e., Legal, Engineering, HR, product management) understand the implications of FOSS in the context of the company's business and fully support the company's strategy around FOSS.
- Communication between the legal and engineering teams around the use of FOSS is open and frequent.
- The company has a defined policy on incoming software that covers both proprietary and open source code in light of the company's culture and business. This policy provides clear guidelines to employees (and contractors) about when and for what purposes different FOSS can and cannot be used and when legal review is required or not. The policy also provides for analysis of the patent and other intellectual property value of technology prior to its release outside the company, including a formal approval process for permitted employee participation in, or contribution to, external FOSS projects.
- When in-licensing software and throughout the development and production process, the company consistently asks and considers questions such as: are we bringing this software in for evaluation purposes only? For internal development and testing? Do we intend to modify this particular software? How do we expect to link it (i.e., statically or dynamically) or otherwise

[1] Some useful articles include "Open Source Intellectual Property and Licensing Compliance: A Survey and Analysis of Industry Best Practices" by Brian Fan, Andrew Aitken, and John Koenig (2004) and available at http://www.olliancegroup.com/opensource/; "Best Practices for Managing Software Intellectual Property", a whitepaper published by Black Duck Software (2005) and available at http://www.blackducksoftware.com/resources/index.html; "Maximize Open Source Benefits with Sound Policy" by Michael Goulde with John R. Rymer, (2005) and available at http://www.blackducksoftware.com/resources/index.html; and "Best Practices for Securing Your Software Intellectual Property Integrity" (2005), a whitepaper published by Palamida, Inc. and available at http://palamida.com/whitepaper/index.html .

combine it with other company and third party software? Are we acquiring it for internal operations purposes only? For inclusion in a product that will be licensed to end users (or via resellers or OEMs) on a royalty basis? Will it be used to run a back-end server supporting an ASP or other kind of services business? Do we need "downstream" rights to sublicense or further distribute? The answers to these questions help determine what inbound license terms are and are not accepted by the company.

- The company has in place a process whereby all vendor/third party code receives an appropriate level of technical and legal review for compliance with license terms and license compatibility throughout the software development and production process, including evaluation/testing, development, production and distribution (including "downstream" licensing or redistribution).
- The company has clear policies covering the use of FOSS code by outsourced developers and other vendors, including representations and warranties required from such vendors. These have been communicated to all company vendors as well as to the company's purchasing and contract management organizations in advance of contract negotiation.
- The company's IT resources are engaged so as to create a central repository for third party and "home-grown" code, and the company has purchased, or designed, a software tool to conduct and maintain a dynamic inventory of all software used by the Company. This software tool tracks where and how the software is used, when and how it is modified, applicable in-bound license terms, and assists the legal team in assessing license compliance and compatibility.
- The company has designated and publicized internally those individuals responsible for handling questions and approvals regarding the inflow, use and outflow of software code.
- The company has an intellectual property/software review board that consists of legal, engineering, human resources and product management staff that meets regularly and that is responsible for implementation, periodic review and enforcement of the company's FOSS policy.
- Company policy and accompanying practices are periodically reviewed and consistently enforced over time.

Some companies are much closer than others to achieving this ideal scenario. In my experience, though, putting these kinds of "best practices" in place is far easier said than done. We now know, or think we know – what to do. The question we face at this point is: how do we actually do it within the particular context of our client companies and given the realities of FOSS?

GETTING WITH THE PROGRAM MEANS OPENING YOUR MIND TO FOSS

Chances of implementing an effective policy are increased when we as lawyers are willing to learn about FOSS and adapt to the particular cultures of our client companies and the FOSS Community. We need to relinquish our role as "gatekeepers" that control the influx of intellectual property into our organizations, and recognize that FOSS requires more open and collaborative lawyering. Rather than working alone in our offices or cubicles, we must walk down the hall (or fly around the globe) to work side-by-side with, and manage across, functions.

Here are some more specific suggestions for how lawyers can effectively implement FOSS policies and practices for their clients:

Recognize How Open Source Licensing is Similar to Proprietary Licensing.

One of the reasons we know that we can create plans, policies and processes for managing the inflow (and outflow) of FOSS to and from our organizations is that we have done so – or should have done so – before. Much of the analysis relating to FOSS code, and many of the "best practices" discussed above, are the same – or should be the same – as those relating to proprietary code. In considering what third party code they will bring in, whether proprietary or FOSS, commercial entities should be thinking about how that code fits within their overall business plan and strategy, future product line, distribution and revenue model. With that context in mind, companies should then carefully select those vendor products required or useful to implement that particular product development and marketing strategy. Companies should be doing a technical review and evaluation of potential incoming third party code for suitability within their own product; and reviewing (and/or negotiating) the applicable license to be sure that they have the rights they need throughout the evaluation, development, production and distribution processes. And, of course, every company should have systems in place to monitor and enforce compliance with those license terms, and to track software acquisition and development, throughout all phases of technology development, production and release.

Given that we have done at least some of this before, why does FOSS seem so different and so scary? Because there are important cultural differences between the world of proprietary software and FOSS that lawyers need to acknowledge and understand if they are to be effective working within this new paradigm.

Recognize How Open Source Licensing is Different from Proprietary Licensing.

Licensing Isn't Just for Lawyers Anymore

Whereas lawyers used to be company "gatekeepers" without whose approval no software could come in the door, now engineers can – and do – surf the web and freely

download to their computers at work and at home a wide variety of software, which comes in as both source and object code, and which is governed by a huge potential variety of license terms. When software comes in as source code, developers' ability to modify and create derivative works from it increases, which means it is easier to commingle with proprietary code – including a company's "home-grown" code and third party code. The consequences of such commingling – particularly with respect to FOSS code governed by "copyleft" license provisions – can be severe, such as triggering requirements to make company and other proprietary source code publicly available upon distribution of commingled code.

To be successful as a lawyer in this new world of FOSS, it is essential that lawyers for commercial entities recognize that there is no more ivory tower. Many developers know – and even more <u>think</u> they know – much more about FOSS software and licensing than the average high tech company lawyer. Until recently, engineers have enjoyed a lot of freedom to download, use and play with FOSS without questions or interference from lawyers. They also have been writing licenses, sometimes without any input from the legal profession, and sometimes those licenses include technical restrictions that the average lawyer won't understand at first glance. Many developers have been working with FOSS a lot longer than we lawyers have.

Understandably, therefore, developers may resist lawyers' efforts to intervene and create policies that restrict their freedom to play with the exciting variety of software out there, or to release software they develop on the terms they choose. This places a burden on lawyers to build relationships with the engineering community (and particularly the FOSS Community) as never before. Given the vast array of software available and the sometimes bewildering license terms under which it is made available, lawyers wanting to learn about FOSS will need to ask for help from developers if we are to learn about available FOSS technology and interpret the variety of sometimes quite technical licensing terms.

We can lend valuable advice to our clients on intellectual property rights, crafting new licenses, and the implications of licensing in FOSS to our companies. But we need to demonstrate that we understand FOSS and why it matters if we expect developers to listen to us. Only then can we effectively educate our clients about the implications of FOSS and create FOSS policies and procedures that they will accept and support. Without developer cooperation, even the most carefully-drafted FOSS policy won't work in practice.

<u>There is a Daunting Amount of Software Out There, with a Corresponding Variety of Licenses.</u>

Another reason that the rise of FOSS strikes fear into our hearts is that the amount of software available, and corresponding number of licenses out there to consider, is truly daunting. When I checked recently, the number of FOSS projects hosted on Sourceforge.net was more than 100,000, with the number of registered users listed as

well over one million. [2] And Sourceforge is not the only place where FOSS is found. FOSS and other freely available software can be downloaded from other open source software repositories, commercial proprietary software companies (such as Microsoft, Sun and Oracle), open source projects (such as JBoss and PostgreSQL), university and personal websites, and commercial open source companies (such as MySQL), among other sources. While many of these software projects release their software under relatively well-known and well-understood licenses such as the GPL or BSD license, many of them release software under different license terms that are referred to here as "FOSS" but may not meet accepted definitions of "open source".[3] Some software is downloadable without a clear path to the applicable license terms, or possibly with no license terms at all.[4] The Open Source Initiative currently lists close to sixty "OSI certified" licenses,[5] but many software developers have chosen to issue their software under terms they have crafted themselves or created by modifying existing licenses[6] – and that do not appear on the OSI list.

Locating, much less understanding and interpreting, the licenses that may apply to code that is being brought into your company can indeed be an overwhelming task. This is where use of software audit and tracking tools such as those offered by Black Duck Software and Palamida, Inc. may assist lawyers in finding what code exists within their company and analyzing those findings. However, just as with anti-virus software, one cannot rely on software alone to flag all issues or problems, particularly since new software is being developed constantly around the world and around the clock. These kinds of tools work well when they are used as part of, but not as the only element of, an FOSS company compliance program. Among other considerations, software does not substitute for good legal counsel.

---

[2] http://sourceforge.net/. Last checked on October 8, 2005, when it listed 103,804 Registered Projects and 1,152,363 registered users.

[3] Various definitions of what "open source" means exist. One of the most commonly referenced is that of the Open Source Initiative, which can be found at http://www.opensource.org/docs/definition.php. Readers may also be interested in the discussion of the difference between "open source" and "free software" found at http://www.gnu.org/philosophy/free-software-for-freedom.html .

[4] For example, see http://www.qmail.org/top.html, which states that "you can. . .redistribute qmail for free" but includes very specific limitations on any modification, including getting Dan Bernstein's prior approval "of the exact package that you want to distribute." See http://cr.yp.to/qmail/dist.html. It has been argued that these terms do not constitute a license at all. See Rick Moen's discussion of Dan Bernstein's "license" at http://linuxmafia.com/~rick/faq/index.php?page=warez#djb and Mr. Bernstein's response to Mr. Moen's comments at http://cr.yp.to/distributors.html .

[5] http://www.opensource.org/licenses/ Last checked on October 8, 2005.

[6] See, for example, the gSOAP public license, which includes this helpful introduction: "The gSOAP public license is derived from the Mozilla Public License (MPL1.1). The sections that were deleted from the original MPL1.1 text are 1.0.1, 2.1.(c),(d), 2.2.(c),(d), 8.2.(b), 10, and 11. Section 3.8 was added. The modified sections are 2.1.(b), 2.2.(b), 3.2 (simplified), 3.5 (deleted the last sentence), and 3.6 (simplified)." See http://www.cs.fsu.edu/~engelen/license.html

In the World of FOSS, Many Engineers Are Motivated by Creating Cool Stuff and Getting Recognition for It – Not by Money.

In the world of proprietary software, the technical specifications of software are reserved as trade secret or otherwise protected via patent and other rights. In the traditional proprietary software model, companies have made money by not telling others what they do, and engineers' motivation for creating "cutting edge" products has been seen as largely financial. In the world of FOSS, that changes. In general, FOSS developers care about creating cool software and being recognized for having done that. They may not care about making profits from the code they create but, rather, want recognition for what they have done and how they have done it. Many of them also passionately believe in the cause of FOSS in general and/or that of particular FOSS projects. The opportunity to contribute to the FOSS Community or "give back" to a particular project is of great value. So using traditional incentives such as bonus payments may not be the best means of ensuring compliance by company engineers with corporate FOSS policies. Instead, letting engineers engage with the FOSS Community on company time and using company resources may be a good way to gain support for your policy from engineers. If a company acknowledges developer contributions and achievements and demonstrates the company's support of FOSS, that may encourage developers to participate and collaborate more with implementing FOSS policies and programs. And, for company lawyers, showing a willingness to learn about, and genuine interest in, FOSS and the FOSS Community helps keep communications between lawyers and developers open.

This difference between proprietary software and FOSS can be important in dealing with third party FOSS providers as well. If a company code audit shows that the company is at risk of violating FOSS license terms, simply offering to pay a large chunk of money to the copyright holder/licensor to settle the matter won't necessarily be the answer. Options for remediation may include ceasing use of the software or replacing components, or publishing the source code for the project. These choices, though, can lead to major code rewrites, associated product launch and sales delays, acquisitions that go south, and bad publicity. However, negotiating a separate license or other solution with the copyright holder on reasonable terms may also be an option available to your client. Here again, recognizing developers' motivation for writing code may well help you to obtain the license terms you need without being "taken to the cleaners". Obviously, there are no guarantees that a particular developer won't see your client as a "deep pocket" or an amusing target for public humiliation, but my experience is that showing support for FOSS by earnest efforts to comply with licenses only helps you. In my negotiations with FOSS licensors, the fact that my client really wants to use the FOSS code that developer has written -- and is making an obvious effort to read and comply with his or her licensing terms -- goes a long way toward smoothing the path for negotiation of license or remediation terms that work for the client.

The FOSS Community Is Committed to the Cause of FOSS and Very Outspoken.

In the world of proprietary software licensing, license terms usually are kept confidential. Unless your client company's licensee or acquirer requires that it make representations and warranties regarding incoming code, or otherwise for marketing and publicity reasons, no one need know from whom and on what terms your software code has been obtained. Since license transactions are primarily motivated by revenue, disputes often can be resolved through payment of money. Unless there is a filed lawsuit resolved via litigation, infringement claims and settlement outcomes are not publicized.

Again, in the world of FOSS, that changes. License terms are published on the web and anyone can read what they say. Members of the FOSS Community are committed to the promotion of open source software, communicate amongst themselves frequently, and see as part of their mission promoting the cause of FOSS and helping enforce license terms. Long threads on http://slashdot.org/ are devoted to happenings of interest to the FOSS Community, including notifying others of potential license violations. Pity the company or individual whose name appears in such an accusatory posting! It may be hard to prove innocence when guilt is presumed, and guilt may be presumed in a very public – and sometimes a very nasty – way. Among other outcomes, this kind of publicity can create a public relations nightmare for your client. The forthrightness of members of the FOSS Community comes out through other channels as well. Messages coming through on the Open Source Initiative's license discussion list[7] vary widely in tone, with requests for interpretation or approval of new licenses receiving responses that may range from thoughtful and gentle to outright "flames". It is therefore not surprising that companies and individuals might think twice before wanting to engage.

Recognize that Every Company is Different.

Just as with other corporate compliance programs, there is no one FOSS solution that works for every company. Each company's FOSS program must depend upon the particular company's business plan and strategy, its product line and revenue model (i.e., whether it makes money through royalty-based licensing, an ASP model, professional services, etc.). Perhaps equally as important is the company's culture and the way that engineers and lawyers interact. Some companies are highly process oriented, and the legal department has more "control" over what behaviors are and are not permitted. In those environments, it may be easier for the legal team to impose and implement policies and processes around FOSS. In other cases, the companies were built and thrive on chaos, and counsel's attempt to impose a policy restricting use of FOSS is likely to flounder. I have been told of one relatively large company where the FOSS "policy" is that engineers are encouraged to use FOSS and contribute back to FOSS projects, and they do so only with the loose oversight of the company's FOSS guru. This sounds like every lawyer's nightmare (and I should note that this company is not currently a client of mine!), but it seems to be working in that case. Some of the reasons for this may be that

[7] To learn more or subscribe to this list, go to http://www.opensource.org/licenses/index.php.

this company was built using FOSS; its founders and engineers have close ties to the FOSS community and are very familiar with FOSS; and the FOSS community views it as friendly and supportive of the FOSS Community. So the company culture itself encourages cooperation with the FOSS Community and, correspondingly, issues of FOSS license compliance are more likely to be worked out informally before they escalate out of control.

The point of this example is not to say that companies should just make friends with the FOSS Community and leave it at that. Rather, it is to emphasize that each company's policy and management of FOSS depends heavily upon the culture and business approach of the company itself. Companies that rely upon royalty-based licensing of proprietary software products need to be aware of the influx of FOSS throughout their product development and production processes – and their FOSS management practices should include careful code audits and reviews throughout the software development and release process. Companies whose business is ASP-based may not need to be as vigilant – although they should not relax into thinking that they can bring in GPL'd or other "copyleft" software and simply use it for back-end server operation without worrying. For one thing, these companies should be alert to the question of what might be considered "distribution" within their own entity – if multi-national corporations have affiliated companies that use "back-end" software developed at one affiliate to provide consulting services or run an ASP offering at a sister company, is that a distribution? Does this analysis change if the affiliated company is a joint venture that is only partially owned by the parent? What about the company's use of outsourced consultants to develop its back-end server software? If software is sent back and forth between the outsourcer and the client company during the development process, is that distribution? Plus, there are other considerations: some licenses may explicitly provide that offering software on an ASP basis equals distribution, thus triggering the reciprocity requirements of that license.[8]

Turn Resistance into Collaboration.

As I have described, software developers have enjoyed a great deal of freedom in recent years and they don't want to lose that freedom now. They have been able to go on the web, pull down software that they think looks interesting, and play with it – without anyone from legal asking bothersome questions. If your client is lucky, its engineers understand the implications of downloading FOSS software and have read – and care enough to abide by – the terms of both their employment agreements and the applicable license before they do so. However, even the best-informed and best-intentioned developer may forget about license terms that permitted testing and evaluation, but

[8] See, for example, Section 1.7 of the RealNetworks Public Source License v. 1.0, available at http://opensource.org/licenses/real.php. This Section provides that ""Externally Deploy" means to Deploy the Covered Code in any way that may be accessed or used by anyone other than You, used to provide any services to anyone other than You, or used in any way to deliver any content to anyone other than You, whether the Covered Code is distributed to those parties, made available as an application intended for use over a computer network, or used to provide services or otherwise deliver content to anyone other than You."

prohibit modification or distribution, by the time they start incorporating the third party code and using it to develop product. Most of them, in my experience, understand the "copyleft" implications of the GPL and are sure that they can effectively navigate around those.[9] They have been working with FOSS for a long time and don't think they need lawyers telling them what to do, nor how to do it. The challenge here is persuading the engineering staff that you want to work with them in managing the influx of FOSS, and that you are not out to create roadblocks unnecessarily.

Face-to-face contact can be very important here. If you take time to walk around the engineering departments and/or set up brown-bag lunches or other meetings with engineers to discuss FOSS issues, you start dispelling the notion that lawyers are the distant enemy. Being an effective lawyer in the world of FOSS requires a high degree of technical knowledge; start asking questions and asking for help, and you are likely to learn a lot. Again, if you demonstrate that you genuinely are interested in technology being brought in and developed, and that you are open to the company's use of FOSS and engagement of the FOSS Community, you likely will find that engineers are more receptive to your involvement.

Company management also may resist imposition of an FOSS policy and management process. The reasons for this resistance vary: small, start-up companies may think it is too expensive or too complicated to institute an FOSS management process. Their engineering VPs tell the other executives that "engineering knows all about FOSS" and "not to worry". The CEO and CFO want to limit legal fees and focus all resources on what they see as mission-critical operations and selling product only – not on lawyers creating new processes that may slow things down, nor on costly software audit tools. Here, education of the executives about the implications of FOSS for their business is critical. Still, you may be fighting an uphill battle. It may take a triggering event, such as a proposed merger or out-licensing agreement where the company is expected to provide comprehensive "no FOSS" warranties – and finds after diligence that it can't -- before the executives pay attention.

Large companies, too, have their own reasons for resistance. Here, the legal and engineering staffs may be large and may operate remotely or independently from each other. The executive team may be complacent in thinking that, since the company is running well and making money, engineering and legal are on top of any issues and there is nothing to worry about. Ironically, I see large companies start to pay attention when they are licensing in software from – or purchasing – smaller companies that they perceive as being disorganized or badly managed. If they do begin to ask questions and perform source code audits relating to the target companies' software, they start

[9] Provided your engineers truly do understand the requirements of the GPL, software entering your company under that license may not be your biggest concern. Larry Rosen discussed the fact that "the GPL is not a disease" in his 2001 article "The Unreasonable Fear of Infection", available at http://rosenlaw.com/html/GPL.PDF. What may be more likely an issue is engineers' – and others' – complacence around the terms of the GNU Lesser General Public License (available at http://opensource.org/licenses/lgpl-license.php), which is often viewed as being a "safe" license. In fact, the LGPL contains "copyleft" terms similar to those of the GPL, to which lawyers and their clients should be alert.

becoming aware of how extensively FOSS is used "out there" and of the pitfalls that may be involved. Company engineers doing the review of the target company's code may drop mention that the client company happens to use FOSS software just like that found at the target. Under such circumstances, executives may be more open to looking under their own hoods. Again, educating the executive team about the implications of FOSS – and the likelihood that there is quite a lot of it in their company already – is important. Executive buy-in and support for an FOSS policy is key to making it work.

Understand that this is a Project for the Whole Company.

Getting executive buy-in becomes all the more critical given that implementing best practices around FOSS is necessarily a cross-functional effort. Effectively managing FOSS in the enterprise requires the involvement of multiple functions – not just legal and engineering. For example, you will want to involve the Human Resources department in restructuring employee agreements to address employees' authorization (or not) to bring in FOSS code and to contribute code back to FOSS projects. Some companies have policies that explicitly permit engineers to spend a portion of their work week engaged in outside projects; often these are FOSS-related. It also will help to have HR's cooperation and support in sponsoring employee training on FOSS issues at employee orientation and other ongoing training sessions.

Another group that should be engaged in managing a company's use of FOSS is the Information Technology group. Depending upon what your FOSS policy provides, you may need IT's cooperation in setting up consolidated code repositories, appropriate firewalls and internet access that permit (or not) large data files being moved in and out of the company. Employees engaging with the FOSS community likely will be using email, instant messaging and other means of communication to do so. They may be working on FOSS projects on company equipment, both at the office and at home. IT and HR should be involved in formulating and enforcing rules around employees' use of company communications tools and equipment for purposes of engaging with the FOSS community. And IT should be involved in decisions around the purchase and implementation of software audit and tracking solutions such as those offered by Black Duck Software and Palamida.

Other groups may need to be involved as well. If your company offers consulting or other software development services, your education and training programs should include the professional services staff as well. You need those employees to understand the potential implications of their use of FOSS software when working with customers' code, as well as the company's own products. Sales and marketing people need to understand to what degree and how your products incorporate FOSS, so that they can anticipate and respond to questions from customers. Purchasing and contract management staff likewise should be familiar with, and equipped to enforce, company policy and requirements around vendors' use (or non-use) of FOSS in any products or services coming into the company. If the company chooses to engage with the FOSS community by funding FOSS projects or organizations, or by contributing intellectual property to FOSS projects, its public relations team will necessarily be involved.

In short, don't expect to implement FOSS best practices alone.

CONCLUSION

Despite protestations from the FOSS Community to the contrary, there are valid reasons that "Fear, Uncertainty and Doubt" linger around FOSS. FOSS is indeed different than proprietary software, and it involves distinct benefits, risks and cultural challenges. Despite that fact, I encourage lawyers working in this area to lose the "Fear" in "FUD". We do know what to do, and we have done much of it before. Our likelihood of success, however, will increase if we don't see ourselves or the legal function as working in isolation. If we are to be effective, we will need help from developers – and, indeed, from other lawyers – who have more in-depth knowledge of FOSS technology and licensing terms than we do. When we are open to asking for help and to learning from the FOSS Community, the FOSS Community likely will be more open to working with, and learning from, us. In sum, Open Source requires Open Lawyering.

And the FOSS Community in fact has much that it can learn from the legal community. I believe that lawyers working with the FOSS Community can help extinguish "Fear, Uncertainty and Doubt" through our participation and, indeed, that we have a responsibility to do so. In this paper, I have discussed the challenges of navigating the vast array of FOSS technology and licenses, many of which contain confusing or conflicting terms. I also have touched on the passionate, almost religious, commitment of many members of the FOSS Community to their cause. While those aspects of FOSS make entering the terrain of FOSS both challenging and exciting, they also work against "mainstream" acceptance and support of FOSS. As lawyers, we are uniquely qualified to offer much-needed perspective around FOSS – including through providing objective analysis and interpretation of license terms. We can help clarify and define the legal rights and responsibilities of the FOSS Community and users of FOSS and, by doing so, reduce "FUD". We also can help craft licenses that are clear and enforceable across jurisdictions. And we can assist commercial enterprises to see how they can bring FOSS into their organizations without risking their core businesses.

Successfully implementing FOSS programs for our companies doesn't just serve our clients' immediate and financial interests; it supports the FOSS Community by reducing the "FUD" around FOSS. And, by collaborating with and supporting the FOSS Community, we don't just increase the chances that our FOSS corporate compliance programs will work, we create an opportunity for lawyers to participate in defining the future of FOSS and technology as a whole.