A New Wave of IP Risks:

How Open Source is Changing IP Risk in the Software Supply Chain

ABOVE THE LAW

FOSSA

Table of Contents

ABOVE THE LAW

FOSSA

I. Introduction

Open source software has been integrated into nearly every industry and sector today. According to a 2016 survey, approximately 90% of today's organizations report using open source software.[1] That percentage has almost certainly grown since. One likely reason for open source's boom in popularity is the distinct cost savings it gives companies who use it.[2] The use of open source software is now so widespread that many companies are unaware of how and where it is used, and would be unable to identify all their open source code if asked to do so.

As Mark Radcliffe, a partner in the Silicon Valley office of DLA Piper specializing in IP and open source, explains, "virtually all software now has a large number of open source components." While this widespread proliferation is a testament to the success of open source, it also gives rise to unique challenges for businesses, particularly in the area of intellectual property. If a company cannot even find all of its open source code or identify its open source dependencies, they are also likely unable to ensure that they are remaining compliant with open source licenses and protecting themselves from business or reputational risk.

In this paper, we will examine the most common IP risks that arise from the use of open source software today, including copyright infringement, patent infringement, reputational risk, exposure of IP secrets, and the impact on the partner/customer relationship.

II. Brief Overview of Open Source and Where It Stands Today

Open source software is software made available in source code form that can be used with no field of use restrictions, modified or redistributed by anyone at any time. Closed or proprietary software, in contrast, is generally provided solely in object code form so it cannot be modified (source code is rarely made available l), imposing restrictions such as user limits, server limits, or limits as to purpose, territory, or kinds of use.[3]

Complying with software licenses is just as important for open source software as it is for proprietary software. There are over 80 open source licenses formally recognized by the Open Source Initiative, many of which are regularly updated; but, there are hundreds more that can be found online which are not formally recognized and many more being created every day. The GNU General Public License Version 2 ("GPLv2" or "GPL") is one of the most widely used open source licenses today and, therefore, the focus of much discussion involving open source compliance. It is estimated that some 16 billion lines of code are licensed under GPLv2.

Understanding the implications of all of those licenses on the open source code that companies want to implement is a full-time job for open source lawyers. For companies, developers, or even attorneys without long experience in open source software, it is nearly impossible:

> "In practice, even determining how any particular piece of software is licensed is not straightforward. Some software packages aren't marked with any license at all. There might be licensing information on the project's homepage instead. While many software packages have a COPYING or LICENSE file indicating the package's license, many also have additional license information in other files in the package. It's not uncommon to see packages

1 https://siliconangle.com/2017/06/02/enterprise-open-source-adoption-skyrockets-linux-addresses-ease-use-guestoftheweek-devnetcreate/
2 https://www.eweek.com/servers/open-source-software-gives-competitive-advantage-gartner-survey
3 https://www.globalpatentfiling.com/blog/ground-breaking-decision-open-source-software-versata-software-case?utm_source=Mondaq&utm_medium=syndication&utm_campaign=LinkedIn-integration
4 https://opensource.com/law/14/12/gplv2-court-decisions-versata

> with 10 or more licenses and finding all of them is either a long manual effort or involves automation. Additionally, it's not uncommon to see a package licensed one way within the package itself but for there to be either additional or conflicting terms on the project homepage or an idiosyncratic explanation with regard to how that project interprets its chosen license."[5]

As Radcliffe explains, many people and companies were initially opposed to open source, but "it's become the dominant form of software development." Widespread use, however, does not mean there is clarity in the field. In fact, the reality is quite the opposite, according to Radcliffe:

> "What I think is interesting is that, even though it's now the dominant form of software development methodology, many fundamental legal questions remain uncertain. Questions as simple as 'What's the scope of the General Public License version 2? What's a derivative work? Does it really include collective work?'"

However, "despite the great deal of legal uncertainty, the tsunami of open source has just washed over all software development," he concludes.

Courts in the United States and abroad have ruled that open source software licenses are binding licenses, shoring up the basic foundation of open source licensing's efficacy and validity. The idea that violations of open source licenses constitute copyright infringement, which therefore also subjects infringers to the wide scope of damages made available under the Copyright Act, including statutory damages, is no longer controversial. Although there have been relatively few lawsuits filed in the United States involving open source and relatively little said about the scope of GPLv2 in particular, the existing decisions point to the possibility of cases with significant legal risks, particularly of injunctive relief and possibly damages. Companies are proposing aggressive and novel interpretations of various open source licenses, some of which may not line up with conventional interpretations of those licenses, have the potential to upend many companies' compliance practices and force them to reconsider their compliance approach and even, potentially, revise their products.[6]

According to Radcliffe, there is also a lack of widespread commercial agreement on how open source risks should be treated. The result is that "everybody in the supply chain needs to make their own assessment of what's in the software that they're getting, and whether the software is compliant with the licenses." Radcliffe describes this as a tug of war between licensors who offer lower prices because they include open source software (frequently with poor knowledge of the licenses used within the software and even less certain compliance with license terms), and licensees who believe those licensors should bear the responsibility for the open source software they choose.

The lack of commercial agreement as to how that tug of war should be handled has created substantial gray areas when it comes to remedies for open source compliance violations. Even more significant than possible litigation for many companies is the serious potential for reputational damage that may stem from being a bad actor in the open source space.[7]

5 https://katedowninglaw.com/2019/06/17/part-1-the-dark-practice-of-free-open-source-software-law/

6 https://opensource.com/law/14/12/gplv2-court-decisions-versata

7 See Section IV.B. of this report.

III. The Interplay Between Open Source and IP Risk: Understanding the Challenges

Significant IP risks exist in the realm of open source, including copyright and patent infringement. However, before companies begin to parse the legal risks associated with open source software, it can be useful to understand the broader open source landscape and the genesis of the disputes that tend to arise.

As Kate Downing, a California lawyer specializing in IP and open source, explains, the interplay between open source software and IP risk is a legitimate concern. However, companies should not just focus on the specific types of risk. Instead, these risks should be analyzed in terms of the types of litigants involved.

In her view, the largest legal risk is posed by litigants who license their software under both open source licenses and commercial licenses and/or companies who have a suite of open source software and commercial software. Often in these instances, it is very difficult to understand the nature of the dual licensing and the open source license which pose compliance problems, such as the Affero General Public License version 3 and, therefore, whether or not a given company is in compliance. Radcliffe notes that such situations are common for companies employing "open-core business model[s], where you basically have an open source community edition of the software, and then a proprietary version, which generally has additional functionality."

Too often, companies will assume that software or code is free to use because a particular company has a reputation for open source development, when that is not in fact the case. This lack of clarity is a significant issue, and it is compounded by the fact that the companies who hold the software are in business to make money. "They have every incentive to pursue violators and get paid for licenses because that's how they make their money," Downing explains.

Most of open source doesn't fall into this high risk category, though, according to Downing. "The vast majority of open source that people use comes from projects that are not incorporated -- they're just a single person or a group of people [building and maintaining the code], and they don't have a legal entity. So, number one, they're probably not copyrighting any of their code. They're also probably not filing patents on their code and don't really have any money or interest to sue you."

Downing summarizes the overarching challenges presented by open source software:

> "If you're really wanting to talk about challenges, per se, I think the challenge is that 90% of what a company ships as part of a product is now open source and third party code. It's only the very tip of the iceberg of what they ship that is actually [proprietary]. And so assessing risk and understanding the IP world has to begin with acknowledging that the vast majority of what you ship is not yours. In fact, not only is it not yours, but you can probably never really get a handle on everything that goes in there. With the advent of package managers, we've got really easy ways for companies to add open source dependencies, and we've automated the ways that those dependencies can add additional dependencies.
>
> Even an engineer who pulls something in may not even know that they're pulling in these 10 other things that this package also depends on. And that's your frontline person. That's the person who knows better than anybody else. So you really have to understand that there's no absolute knowledge in

this area and there's really no way to get it. Even the best tools are imperfect. Even if you do a very, very good job of scanning and you find every single piece of open source and third party code that you include in your product, going from there to making sure that you comply with every single license that there is, is even harder."

The inability of companies to ever completely know the full universe of the open source software they use begs the question of how companies can hope to avoid IP and business risks. Downing advises:

"I think the approach to this has to be to take reasonable measures step-by-step and, in general, to do things that are industry standard. That's my advice to my clients—be a little bit cynical. You don't have to outrun the bear. You just have to outrun the other people with you. You should be scanning your code. You should be putting together attribution files. There is a basic level of compliance that you should be doing because that's what everybody else is doing. But there are some things that nobody is doing because it's extraordinarily expensive or technically difficult. The goal is to be in reasonable compliance and weigh the risks and benefits of that compliance."

The reality, explains Downing, is that full compliance is an effort that may take several years to achieve. Even after a few years, companies will still have gaps, and compliance will not be perfect. While this concept may be daunting, she believes it is an important one to convey so companies begin to understand their potential IP risks. Companies need to be informed so that they can minimize their risk. As Downing summarizes, "Your risk is always going to be a sliding scale. You're never going to reach perfection. The real challenge is understanding that and still committing to doing your best."

Understanding IP risk in open source necessarily requires understanding the potential consequences for bad acts. "Ultimately, lawyers advise their clients based on a risk analysis: if we don't get this right, what is the likelihood we will be sued and by whom? Here, lawyers start looking at who the entities enforcing open source licenses are, what types of noncompliance they are targeting, who they are targeting, and what their goals and motivations are."[8]

Downing frames the core issue: "It's most important to think about where risk is coming from rather than the nature of the risk. It's not, "what can I be sued for?" It's, "who am I going to make angry?" And a lot of times when you're looking at things and you're making decisions about which way to go, that's a very practical risk assessment." It's frequently not, as she notes, a legal analysis so much as a common sense one.

8 https://katedowninglaw.com/2019/06/17/part-1-the-dark-practice-of-free-open-source-software-law/

IV. Specific IP Risks in the Open Source Software Supply Chain

A. Copyright Infringement

While the initial copyright concerns about copyright law in open source software about whether such software might include copyrighted code from third parties who may later sue the project or licensees of the project, this concern quickly became a non-issue due to a lack of interest in enforcement. The focus in copyright law quickly shifted to copyright license compliance, which continues to be a dominant issue in open source software enforcement today.

1. Copyright Diligence Risk

The potential IP risks posed by open source software have changed as its use has become more prevalent. When companies first started using open source software 20 or 25 years ago, there was a level of discomfort with using free software that lacked the backing of a commercial vendor. For that reason, Heather Meeker, a partner in the Silicon Valley office of O'Melveny & Myers specializing in copyright and open source, explains, many IP lawyers at the time considered open source software extremely dangerous to use:

> "The reason they thought that was mainly that, when you're running an open source project, anybody can contribute to it and you cannot possibly do copyright diligence on what they're contributing. You can engage in some best practices, but you don't necessarily know where everything is coming from. Therefore, at that time, most lawyers were actually counseling their clients not to use open source software."

Ultimately, though, the concern over copyright diligence presenting a risk fell by the wayside. Meeker explains why:

> "The kind of contributions that people usually give to open source projects are not the kind of IP that generates rights that people care about enforcing. If I'm running a project, and somebody steals a few lines of code from somewhere and contributes it, the incentives just don't exist to try to find and address that problem. Now, of course, it's not like it can't happen or it never happens. It's just extremely rare for it to happen."

The open source code, Meeker explains, is essentially a free good, like a public road, whereas the software running off that code higher up the stack is more of a commercial good, like the cars on the roads. It would be too difficult to charge everyone for the roads, but no one expects the cars to be free. Therefore, while there may be numerous instances of copyright infringement in underlying open source projects, they are not the kinds of copyright risks that rise to the level of enforcement. For that reason, any risk originally identified in the early days of open source lacked urgency in the larger analysis.

2. Compliance Risk

The greater copyright risk presented by open source software is commonly referred to as compliance risk. Nearly every company today uses open source software, ingesting other people's open source software and using it in their own organizations. When a company does so, it needs to ensure that it is complying with the relevant open source licenses.

The bulk of IP law in the open source realm today relates to license compliance. Companies run audits on the codebase of a piece of open source software to determine what makes up the software and what licenses cover it. As Meeker explains, there are two categories of open source licenses:

"There are the permissive and copyleft licenses.The permissive licenses are easy to comply with because they basically say, 'Here's code, do whatever you want with it,' and they have a requirement that, if you redistribute it, you have to include a copy of the license. You can take code under permissive licenses, put it in proprietary products, or put it in other open source projects. It's difficult to violate those licenses unless you have completely ignored compliance altogether and you're not delivering your notices.

Compliance difficulties arise with the copyleft licenses. Copyleft licenses have significant additional conditions attached to them. If you redistribute the code, you must redistribute it under the same licensing terms. A great deal of complexity goes into what gets captured by that requirement."

The number one license creating open source compliance risk, according to Meeker, is the GPLv2. The GPLv2 is a copyleft license, the most common open source license for projects and has the most aggressive "copyleft" terms.

3. Best Practices for Open Source License Compliance and Penalties for Noncompliance

The first step in complying with open source licenses is knowing what software is in the products you use. While that may sound simple, the task is complicated by the fact that today's engineers often source large amounts of software from the web with less-than-optimal diligence paid to keeping track of what it is and where it comes from. As Meeker explains, the difficulty of the task depends on how much code has been sourced from the web:

"If somebody takes an entire library of code and uses it, that's reasonably easy to identify. If they cut and paste two or three lines, it approaches a nearly impossible task to figure out. So, if the engineers are not extremely circumspect about where they're getting their code from and keeping records, then you don't know what's in the codebase. And if you don't know what's in the codebase, you cannot possibly comply with the licenses because you don't know what they are. So the number one question is: What's in the codebase?"

Penalties for noncompliance are most typically not of a monetary nature because most of the enforcement of open source licenses is done by the community. Authors who have released the open source code will inform community enforcers that they've released the code under GPLv2, and the enforcers, in turn, will approach the violators to try to obtain compliance. As Meeker explains: "They don't usually want a lot of money, but that process can be extremely disruptive. While lawsuits and injunctions rarely come to pass, severe disruption of business does." If an enforcer requires immediate license compliance, a company must take resources away from product development and, instead, focus them on compliance, delaying product releases.

However, a contributor to Linux, Patrick McHardy, threatened (and sometimes sued) companies in Germany to make money. He has been estimated to have received over $2,500,000 from about sixty companies. He is frequently referred to as a "copyright troll."[9] Linux has over 14,000 contributors and they may have different views of what is appropriate. While litigation may not be as common in open source matters as in other areas, there have been a handful of significant cases that have shaped the law in this area. Jacobsen v. Katzer, a 2006 case from the United States District Court for the Northern District of California, established that violations of open source licenses should be treated as copyright infringement claims rather than breach of contract claims. Meeker explains:

9 https://opensource.com/article/19/2/top-foss-legal-developments

> "When you violate a license, an open source license, the claim that's brought against you is a copyright infringement claim, not a breach of contract claim. A breach of contract claim can be brought as well, but they're usually not as interesting, in terms of damages and remedies, because copyright has statutory damages, actual damages, disgorgement of profits, and an injunctive remedy available. Whereas with contract law, you usually don't get those things. So the cases are brought as copyright infringement claims under the theory that, if you violate the license, you are not licensed and therefore are a copyright infringer. Jacobsen v. Katzer confirmed that. So that was a landmark case. It was the result everyone expected, but nobody had really seen a case coming out of an appeals court until that point."

One of the most significant open source cases in U.S. jurisprudence is Google v. Oracle America. This long-running case originated in the United States District Court for the Northern District of California and is currently pending before the United States Supreme Court for argument in October 2020, It started as an open source compliance case and evolved into what Meeker calls "the copyright war of the century."

Despite the prominence of that case, Meeker points out that many open source cases are brought in Germany (such as Patrick McHardy), which is a very plaintiff-favorable jurisdiction when it comes to open source enforcement. Still, compared to other areas of law, open source sees very few lawsuits. Meeker states, "There have been a handful of cases, but relative to other kinds of IP cases and enforcement of proprietary software, it's a drop in the bucket when it comes to actual court actions."

Downing concurs, stating, "There have definitely been lawsuits, and courts have said open source is a real license, and you can face copyright infringement for violating it. So, I think we know what the risks are. But there hasn't been a lot of litigation around the very detailed nature of open source. No one has ever really answered questions within the United States, such as: What is the scope of GPL or what is the scope of a derivative work ?"

Radcliffe explains why most open source cases do not end up in court:

> "There's a variety of reasons. The first one is the ambiguity. Neither side, frankly, wants to risk a decision because the way GPLv2 was written is very ambiguous. You could see a lot of potential outcomes, which is very problematic. So when you look at the cost of going to court, all the uncertainty, and the potential risk, many people decide it's better to just pay to come into compliance to the extent of the demands and then go on their way."

The lack of lawsuits, however, does not mean that open source violations go unpunished. Monetary damages and court costs are not the only risks stemming from open source noncompliance. Both Meeker and Downing agree that the biggest risk with open source software is not facing lawsuits but, rather, public perception and risk to reputation.

B. Reputation

While copyright noncompliance with respect to open source software might not often lead to steep, or even any, monetary penalties, failing to comply with open source licensing requirements can have significant implications on business reputation. As Meeker explains,

> "While they don't usually want much money from people for the violation, they will threaten to expose you publicly for what you're doing. So it's a reputational risk. Also, a company that has a corporate culture of open source noncompliance risks not being able to recruit people and a reputation as a generally bad business environment, just like any company that has a culture of noncompliance with the law would have. So if you have a company that's dumping chemicals into the ground without permits, it's a similar thing. Nobody wants to work for that company because they don't want to be part of an organization that doesn't care about doing the right thing."

Downing states the risk even more directly:

> "In reality, I think the number one risk for all companies with respect to open source is reputation. And I say that because so many people are involved in the open source world, and so many people are interested in companies that do open source. It resonates with them because it's part of their own identity. If you get a bad reputation as an open source community member, it really will hurt you with hiring. It will hurt you with the media. It can derail some of the partnerships that maybe you had in mind. I can't put a dollar figure on those kinds of risks, but I actually think that they're the biggest ones, especially if you're the sort of company that markets itself as being open source-friendly."

Open source reputation can significantly impact a company's ability to recruit and to work with other companies. Sometimes bad behavior becomes widely public through PR disasters, but other times there is just common knowledge within the tech world that certain companies do not comply with open source licenses in one way or another, Downing explains. If the technology is popular, people might be willing to overlook the reputational issues, but as soon as a rival company creates a competing technology that is more compliant, many users will quickly make a switch, she explains, "if for no other reason than their legal department tells them to."

Radcliffe expands on how a bad open source reputation can impact hiring:

> "I think something on the order of 90 or 95% of software stacks now use open source, and the more recent software stacks tend to be [more than] 80% open source. And a lot of programmers, particularly relatively young programmers, think of contributing to open source projects as almost a right. It's something they've all done all along, and they want the ability to continue to contribute in a place that is friendly to open source. So in the competition for talent, having a good reputation in the context of open source is extremely important."

Reputational risk, however, does not exist entirely separately from the litigation risk. In fact, as Radcliffe explains, being embroiled in litigation over open source issues can hurt a company's ability to do business in the software market: "In this interconnected world, people's products depend on multiple levels of third parties. There's also the reputational issue of 'If I'm Company B, do I really want to integrate software from Company A into my product if Company A has a poor reputation for open source license compliance? Even worse, do I want to take the potential litigation risk of Company A's poor compliance? (See the description of the Versata case below where all of the customers of a licensor were brought into litigation with an open source licensor).

C. Exposing IP Secrets

The potential for exposing IP secrets exists any time a company has to release source code in order to comply with an open source license. Meeker explains, however, that exposure is not the model result:

> "It is one way to comply, but it's not usually the avenue anyone takes. What they usually do instead is rewrite code or remediate it some other way. So, yes, there is a risk. When you violate an open source license like GPL, you have a choice of either releasing all the source code for the program under GPL or ceasing using the GPL software. People usually choose the latter route, which means they've got to re-engineer their product."

Some companies, however, do choose to risk exposing secrets. In one famous case involving Cisco and Linksys routers, Linksys, early on in the process, ended up laying open a lot of its code. After Cisco bought Linkysys for $500 million in 2003, complaints surfaced that Linksys was violating the GPL by failing to provide source code for some of the router code. The Free Software Foundation intervened, threatening to enforce the GPL's requirements. Linksys eventually released the source code in question a few months after FSF demanded that they do so.[10]

In a more recent example, automobile manufacturer BMW delivered 950 megabytes of open source software code to a private citizen in order to comply with the GPL. After noticing a reference to GPL-licensed code in the onboard software of BMW's i3 electric car, writer Duncan Bayne requested the source code. BMW originally declined to provide it, but changed course after Bayne posted the parties' correspondence online and it went viral within the open source community.[11]

Meeker stresses, however, that companies always have a choice whether to expose IP secrets, saying, "There is a risk of exposing secrets. But that risk doesn't arise because a court orders you to do it. It exists because you make a decision that exposure is a better way of proceeding than re-engineering the product." As she further explains, no court can or will order a company to lay open source code because of an open source violation.

Radcliffe concurs, saying, "Basically, once something is open source, there's no trade secrets in the source code. So exposing trade secrets is mainly something that people think about if they consider whether or not they want to open source a product."

D. Impacts on the Partner/ Customer Relationship

A company's reputation when it comes to open source compliance can have significant impacts on its relationships with customers and partners. Radcliffe believes, "For people who license software, how companies treat open source has now become one of the biggest issues, in part because there are not very good standards right now."

The potential impact of noncompliance on those relationships increase as companies get bigger, Meeker explains:

> " When you're a small company, you can get away with a lot of noncompliance of various things, including open source, without it hurting you very much. But once you become a serious company, you have to have compliance processes. For example, if you want to sell a product to a customer, the customer these days may very well ask you to disclose what your open source usage is and will definitely ask you to undertake an indemnity for any violation of open source licenses that happens in the course of you providing the product. If you can't show that you have been responsible about compliance, customers may very well refuse to deal with you."

10 https://linuxinsider.com/story/open-source-and-the-legend-of-linksys-43996.html
11 https://www.theregister.com/2016/03/30/bmw_complies_with_gpl/

Similar to the business relationship risk, open source noncompliance can negatively impact a company's ability to obtain funding or complete M&A transactions. According to Meeker, "when you go to get an investment round or you go to get an exit, your open source position will be examined. At that time, an audit will be done and they'll look into the processes you're using internally. If you can't pass muster on those elements, then it's going to slow down deals, devalue deals, or occasionally break deals."

As Meeker further explains, the most common result when there is rampant open source noncompliance in a corporate deal is that an audit is performed, problems are identified, and the company must figure out remedies if they want a deal to close. Those remedies can be included in post-closing covenants or as closing conditions, depending on the degree of the noncompliance or the risk appetite of the buyer or investor.

While noncompliance at early-stage companies may go unnoticed in many circumstances, "these issues definitely come home to roost when you do customer sales and when you do corporate transactions," Meeker explains. "A lot of companies can fly under the radar before then, but they won't be able to do it once they have a serious business."

E. Patent Infringement

Another key IP risk area for open source is patent infringement. As Downing explains, open source patent infringement cases give rise to another category of potential litigants:

"The people who may be suing for patent infringement may not be the authors of the open source software that you're using. They may not even be the company who created the open source software that you're using. An entirely different third party that you've never heard of who owns a patent may come forward and sue you for using a particular piece of technology."

This risk, she notes, existed before open source. People buy and sell patents all the time, and patents exist on things that no one knew were patented or even imagined were patentable. For that reason, patent trolling and the risk of patent suits has always existed. With open source software, there is always a potential for patent infringement if unknown third parties hold a patent with respect to particular pieces of the open source code.

Patents have long loomed large in the open source discussion, Downing notes, dating back to the late 1990s when Microsoft raised patent issues with Linux. The same situation arose later between Microsoft and Android. While there is no guaranteed way to avoid patent violations, Downing suggests a few best practices to follow: "You could focus more on certain projects than others. If some projects are run by people you know or whose pedigree you know and trust, you can have a little bit more comfort with what they're doing for two reasons. First, if a project is widely used in the industry, other companies may have an interest in assisting you in fending off a patent lawsuit. Second, the project itself is more likely to have resources to deal with the legal risk head on or re-engineer quickly as necessary. Note as well that if an entity does launch a campaign of patent infringement suits, knowing what's in your products is crucial so that you can preemptively remove the allegedly infringing code."

Realistically, though, she notes that the risk of patent infringement likely does not increase when you're using open source software. Innocent patent violations happen on a regular basis even with code that's entirely written in-house.

V. Looking Ahead: The Future of Open Source Litigation

The historic lack of court involvement in the open source sphere may be coming to an end. A collection of recent cases involving Versata Software, Inc. is widely recognized to have changed the legal landscape of open source. In Radcliffe's words, "Versata is a cautionary tale" of what might be to come in open source.

Versata originally sued Ameriprise Financial Services for breaching its software license. In the course of the case, however, it was uncovered that Versata itself was potentially violating the GPL, leading to a series of cases that would become the leading decisions on copyright, patent, and other issues in relation to open source software.[12]

Versata was a provider of software for the financial services industry, and Ameriprise used that software in providing services to a network of independent financial advisors. When Ameriprise used a third-party contractor to customize Versata's Distribution Channel Management (DCM) software, Versata claimed those activities violated the software license and sued in Texas state court. In discovery, Ameriprise ascertained the DCM contained open source software from third parties that was licensed under the GPLv2. According to Ameriprise, Versata violated the DCM license by including open source software, because the DCM license stated it did not include any "encumbered" software. Ameriprise countersued, arguing that the way Versata integrated the open source software into DCM rendered all of the DCM software a "derivative work" that was subject to the GPLv2, making Ameriprise's modifications permissible.[13]

Allegedly, Versata also breached the terms of the GPLv2 when it failed to include the required copyright notices, the text of the GPLv2, and a copy of the source code for the third-party open source software when it licensed DCM to Ameriprise. The dispute between Versata and Ameriprise ultimately led to five lawsuits, including suits involving third parties, that raised patent and copyright infringement issues, among other things. Versata ultimately settled copyright and patent infringement claims with the third party. Ameriprises's counterclaim against Versata for breach of the GPLv2 was remanded to state court on the theory that the GPLv2 imposes obligations beyond the scope of the Copyright Act. On the issue of patents, the court concluded that, if one party commits a patent infringement, that would not impact the rights of other licensees who complied with the patent licenses.[14]

Radcliffe summarizes the cases:

> " The net of all this is that the Versata case is not just about compliance with open source licenses. If you've got a product [that] is a mix of open source and proprietary, which is true of virtually all products these days, there is a chance of it interfering with what you would think of as normal commercial decisions, for example the decision to terminate a license. So there are additional reasons to have an active compliance program."

While the Versata cases leave a number of questions regarding the GPL unanswered, they nevertheless constitute some of the most significant legal guidance on GPLv2 compliance to date. Furthermore, together with the Oracle v. Google cases discussed above, the Versata cases indicate that "the days of open source software free lunches are rapidly coming to an end, and that means enterprises that fail to stick to the terms of open source licenses can expect to be sued."[15]

So what is the biggest issue facing open source software users going forward? In Radcliffe's view, it is how companies

12 https://opensource.com/law/14/7/lawsuit-threatens-break-new-ground-gpl-and-software-licensing-issues
13 https://opensource.com/law/14/7/lawsuit-threatens-break-new-ground-gpl-and-software-licensing-issues; https://opensource.com/law/14/12/gplv2-court-decisions-versata
14 https://www.globalpatentfiling.com/blog/ground-breaking-decision-open-source-software-versata-software-case?utm_source=Mondaq&utm_medium=syndication&utm_campaign=LinkedIn-integration
15 https://www.cio.com/article/2893738/how-2-legal-cases-may-decide-the-future-of-open-source-software.html

are managing their use of open source software. "The Versata cases are an example of why you should know what's in your product and make sure that you deal with it appropriately," he says. "My view is that we're going to see more and more of these commercial disputes where the GPLv2 gets involved."

He further notes that the open source management issue "continues to evolve. You'll find that even companies who are very sophisticated and even companies that sometimes have compliance programs don't get it right. Sometimes they get it wrong in very fundamental ways."

More recently two commercial rivals have sued each other for failure to comply with the GPLv2 in their products.[16] The dispute in Ubituiti v. Cambium highlights the kinds of lawsuits that will potentially dominate the open source legal landscape in the future, and for Downing, strengthens her "beliefs that the highest litigation risks related to open source are from for-profit corporations."

She further explains potentially far-reaching effects of such legal disputes between commercial companies:

> " When for-profit entities litigate, they are often willing to open extremely thorny issues related to open source that have no precedent. They are also willing to argue about it solely for the vantage point of what will benefit them in that particular case, regardless of whether that argument might hurt the open source world as a whole or even their own plans with respect to other products or other elements of that very same product. This kind of litigation, therefore, has really high variability for possible outcomes, even if the plaintiff wins."

A key takeaway from the Ubituiti v. Cambium case, Downing says, is that companies should make sure they are in full compliance with all open source licenses before they head to litigation. "If they don't," she explains, "those issues could not only end their litigation as it did here, but they may also be opening the door to allegations or lawsuits from third parties demanding compliance now that they have made their OSS usage very publicly visible. Here, they managed to attract the enforcement of the FSF despite their own case being dismissed."

The open source world is likely to see more cases like Ubituiti v. Cambium in the coming years. In the meantime, to fill in the gaps in guidance provided by the case law to date, two high-profile open source organizations involved with enforcing the GPLv2 have provided comprehensive guidelines for open source compliance: Software Freedom Law Center's Guide to GPL Compliance 2nd Edition[17] and Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide.[18]

For companies that want to avoid open source risks, a focus on active compliance is key. The question of open source compliance is getting more attention than ever, so it is critical for companies to do their due diligence and monitor their open source activities.

VI. Conclusion

Given the widespread prevalence of open source software today, companies need to commit to being aware of both how they use open source software and what the legal and business implications of that use might be. This includes understanding what open source software is included in their products, as well as creating a policy for managing open source software and understanding the obligations that come with it.

16 https://blog.j2sw.com/equipment-2/cambium/ubiquiti-vs-cambium-the-legal-battle/
17 http://www.softwarefreedom.org/resources/
18 https://copyleft.org/guide/