

开源软件法律风险分析与防范

李斌 杜涛 钱岭

(中移(苏州)软件技术有限公司,江苏苏州 216153)

摘要:随着“开源软件”生态链的发展演进,开源软件数量和影响力均不断攀升。由于开源软件成本低、升级快的特点,越来越多的企业选择使用开源软件。但“天下没有免费的午餐”,在形形色色的开源许可证的约束下,开源软件的使用者们并非想象中的免费和自由。不恰当地使用开源软件,既可能会给使用者造成直接经济损失和负面舆论影响,还可能导致使用者技术流失。本文基于对开源软件及开源许可证的法律分析,结合某产品中的开源软件应用实践,总结出开源软件的使用规则,并提出该产品的合规整改建议。

关键词: 开源软件;许可证;兼容性

中图分类号: D923.4;F204

文献标识码: A

文章编号: 1671-2064(2019)01-0029-04

1 开源软件与开源许可证

1.1 开源软件的基本概念

开源软件起源于上世纪70年代的黑客文化。^[1]1998年,为实现开源软件领域的统一认证标准,Bruce Perens和Eric S.Raymond等人于美国加州发起设立OSIA (Open Source Initiative Association,开放源代码倡议组织),该组织为非营利性的行业自律组织,其宗旨在于推动开源软件在商业中的应用,扩大开源运动的影响^[2]。开源软件具备特定许可证、可获得源代码、享有版权、可自由使用、无任何担保及无许可费等特点。在判断是否是开源软件时,一般以其附带的许可证来进行判断。狭义地理解,开源软件指在经过OSIA认证的许可证下发布的软件。但随着越来越多的个人和企业参与到开源运动中来,部分新兴开源组织或者发布者个人也会编制自己的开源许可证,这部分软件也应该列入开源软件的广义范畴。

从法律权利义务角度来定义,开源软件是指软件权利人在公开源代码时,明确放弃软件版权中的修改权、许可使用权及其报酬权,以此对价换取该软件免费使用者或修改者对后续开发之软件同样公开源代码信息,明确放弃软件的修改权、许可使用权及其报酬权的软件。实质上,开源软件就是在著作权法律框架允许下的知识产权的一种特殊处理模式。在遵守许可证的前提下,任何组织或个人均可以直接获取、访问、使用源代码,对开源软件及其演绎作品进行修改和再发布。

1.2 开源许可证的法律界定

开源许可证是指开源软件随附的、由软件发布者拟定的、并与软件后续使用者或修改者之间签订的、旨在指导和规范软件如何使用的约束文件。开源许可证具有承认版权、发布义务、允许使用和修改、没有担保等共同特性,在

法律属性上,就是合同(即许可协议)。基于开源许可证的合同属性,在使用开源软件过程中,有以下三点值得注意:一是约束性。开源许可证是按照“NO acceptance, No use”的模式发生效力,也就是使用者一旦运行、修改或发布了开源软件,许可协议就会生效,对使用者和贡献者产生约束力。二是自由性。只要不违反法律法规强制性规定,以及上一许可证的规定,软件的贡献者可以自由设定对下一个使用者的许可内容。三是相对性。如果开源软件中包含了多个贡献者,使用者在使用时则应遵守前面每个贡献者发布的许可证的规定。同时,对于违反许可证的侵权行为,每个贡献者只能就自己发布的代码部分进行维权。

1.2.1 常用的开源许可证

如上所述,开源许可证具有自由性。理论上,每个软件发布者都可以在合法的前提下,按照自己的意愿制定自己的许可证。因此,实际存在的许可证非常多,经过OSIA认证的许可证就已超过70多种。但流行的许可证数量不多,主要是GPL、LGPL、AGPL、MPL、CDDL、EPL、BSD、MIT、Apache。上述许可证概要^[3]如下:

GPL许可证(General Public License),为使用最广泛的许可证(GPL 2.0版本)。主要特点有:

(1)允许各种链接、修改,但被链接的整个产品、修改部分均需要开源;但通过pipes、sockets和命令行参数与GPL软件进行通信的情况下,不受开源限制。

(2)有专利免费许可条款、专利冲突不免除义务条款和地域限制条款,来保证专利的共享。

LGPL许可证(Lesser General Public License),LGPL是在GPL的基础上,做了一些宽松的改变,所以也称为宽通用公共许可证。主要变化有:

(1)允许各种链接,动态链接无开源义务,静态链接需

收稿日期:2018-11-04

作者简介:李斌(1981—),男,汉族,湖南新宁人,硕士研究生,研究方向:企业知识产权管理。

要开放与之链接私有软件的文件。

(2) 允许修改再链接到私有软件,但是修改增加的功能实现不能依赖私有软件的数据或功能。

(3) 允许不受限制的使用头文件中的数值参数、数据布局、存取、小宏、内联参数等十行以内的模块。

(4) 仅原则性声明专利应免费许可,无详细规定。

AGPL许可证(Affero General Public License),AGPL同样是在GPL的基础上制定的,其修改目的是为了避开一个GPL/LGPL许可证中被称为Web Service Loopwhole的漏洞,明确了使用AGPL许可的软件与使用者通过网络进行交互(例如舆情服务),也应视为分发行为,需要开放源代码(包括修改部分)。

MPL许可证(Mozilla Public License),使用广泛的为MPL 1.1版本,其比LGPL许可证更加宽松。其关于再发布有以下规定:

(1) 对于MPL许可的代码的修改必须继续以MPL发行,并在发行时附带一份MPL许可证的拷贝,必须提供相应的源代码。

(2) 对于MPL许可的代码的可执行形式可以采用其他不与MPL冲突的许可证发行,但其源代码必须依旧以MPL发行。

(3) 可以将MPL许可的代码或其一部分与其他代码结合在一起形成一个广义作品,并将该广义作品作为一个单一的产品以其他许可证发布,但其MPL许可的部分必须继续遵循MPL的条款。

CDDL许可证(Common Development and Distribution License),使用广泛的为CDDL 1.0版本,与MPL没有本质的不同。对于再发布,原CDDL许可的代码及其修改必须继续以CDDL许可,但允许其可执行代码和广义作品在遵循CDDL的前提下以其他许可证发行,包括使用专有软件的许可证。

EPL许可证(Eclipse Public License),使用广泛的为EPL

1.0版本,替代原先的CPL许可证(Common Public License)。EPL允许任意使用、复制、分发、传播、展示、修改。关于再发布有以下规定:

(1) 对于EPL许可的代码的修改必须继续以EPL发行,而不能改用其他协议发布。除非获得原版权所有者的授权;

(2) EPL许可证下,可以将源代码不做任何修改来商业发布,但如果要发布修改后的源代码,或者当再发布的是目标码时,必须声明它的源代码是可以获取的,而且要告知获取方法;

(3) 可以将EPL许可的代码或其一部分与其他代码结合在一起形成一个广义作品,并将该广义作品作为一个单一的产品以其他许可证发布,但其EPL许可的部分必须继续遵循EPL的条款。可见,商业软件可以使用EPL许可证的代码,也可以修改EPL许可证的代码,但要承担代码产生的侵权责任。

BSD许可证(Berkeley Software Distribution License),使用广泛的为New BSD版本,没有Copyleft的要求,条款约束非常少,几乎接近公共领域。主要特点:允许各种链接、修改,无必须开源义务。无专利条款,即没有对专利进行说明,因此使用时可能会侵犯贡献者的专利权,存在较大专利侵权风险。

MIT许可证(MIT License),比BSD还要宽松。对再发布只有一条规定:原著作权声明和本许可证必须出现在本软件的所有拷贝或实质性部分中。MIT许可证也无专利条款,即没有对专利进行说明,因此使用时可能会侵犯贡献者的专利权,存在较大专利侵权风险。

Apache许可证(Apache License),使用广泛的为Apache 2.0版本。基于传统的BSD和MIT许可证,但是增加了一些重要条款。Apache许可证明示了专利授权。Apache许可证对于再发布有如下规定:

(1) 必须附带一份本许可证的拷贝,必须将原来的源文件中的著作权、专利、商标和归属声明保留在任何派生作品

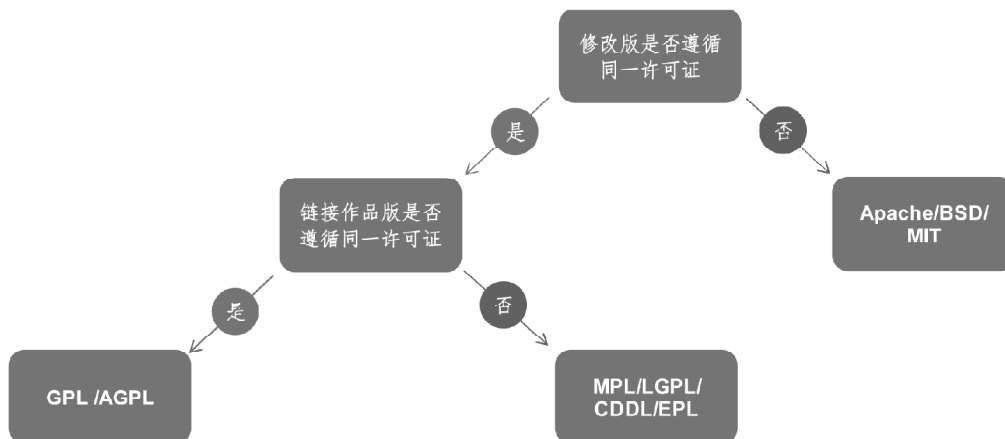


图1 开源许可证间的传染性规则

的源文件中,除非这些声明不适用于派生作品的任何部分。

(2)可以为所做的修改提供再发布者自己的著作权声明和许可证,可以将派生作品作为一个整体用再发布者自己的许可证发布,前提是遵守本许可证的条款。

(3)不能适用授权人的商标;再发布人可以向其用户提供服务和担保等并收取费用,但再发布人必须单独为此担负全部责任。

1.2.2 开源许可证的两个核心问题

虽然,开源软件的发布者给予使用者各项权利,对使用主体、用途等都有限制,使用者可以自由地运行和修改开源软件,但同时也设定了许多义务。在使用过程中,如果不注意,违反了许可证,使用者持有开源软件的相应权利就会终止,并将承担违约责任甚至是版权侵权责任的风险。目前,对使用者而言,最需要关注的就是开源许可证的传染性和兼容性两个问题。

2 开源许可证间的传染性规则

开源许可证的传染性是指使用某种许可证的开源软件模块,在被组合使用的过程中,会对与之组合的其他软件产生影响,以致整个软件也须使用相同的许可证进行发布的特性。这一问题的提出,起源于GPL许可证典型的“Copyleft”特性,因此开源许可证的传染性在GPL、AGPL、LGPL许可证中表现得最突出。

按照图1所示,开源许可证的传染主要发生在对开源软件的修改和链接过程中。具体而言,主要有:

(1)遵循GPL/AGPL许可证的代码,修改/链接版需严格遵循GPL/AGPL许可证发布。

遵循LGPL许可证的代码,修改版需严格遵循LGPL许可证发布。

(2)遵循MPL/CDDL/EPL许可证的代码,修改的该部分会产生传染,即:其MPL/CDDL/EPL许可的部分必须继续遵循MPL/CDDL/EPL的条款。但不会影响整个软件。

(3)遵循Apache/BSD/MIT许可证的代码,修改版可以遵循其它许可证发布。

因此,根据开源许可证中就开源软件的修改、链接源代码所造成的传染性影响关系及规则描述,使用者也可以

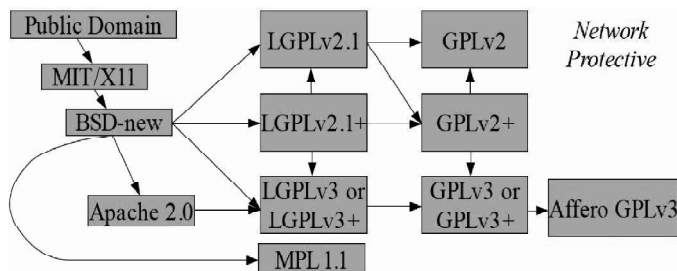


图2 开源软件兼容性

采取一些相应的措施手段,规避许可证的传染。具体方式如下:

(1)使用Linux(GPL许可证)内核的头文件定义,进行系统调用的程序不会被定性为衍生产品。Linux内核的作者Linus Torvalds以及内核开发人员多次澄清普通系统调用为非GPL的作用范围,甚至固化在Linux内核的源码Copying文档中,为Linux用户空间的程序采用非GPL的授权许可证打下了基础。

(2)链接(包括源代码包含,静态链接和动态链接三种情况)使用了其他GPL的类库的程序会被定性为衍生产品。可以通过设置硬件抽象层HAL(Hardware Abstraction Layer)的方式进行规避。无法通过上述措施规避的,则只能通过采用其他许可证的模块替换或自己编写代码的形式规避。

(3)当需使用LGPL的开源代码时,主要通过动态链接的方式进行使用。当需使用MPL/CDDL/EPL的开源代码时,对于MPL/CDDL/EPL许可的源代码的修改必须继续以原许可证发行,而不能改用其他协议发布;可通过链接的方式进行使用。

3 开源许可证间的兼容性规则

开源许可证的兼容性^[4]是指在某一种开源许可证下发布的软件,是否可以集成到采用另一种开源许可证发布的软件中去,即上述两种(或多种)许可证是否均允许上述集成的特征。由于软件的复杂性,在实践中,不同许可证下的开源软件模块的整合是不可避免的,但不同开源许可证条款之间,可能会存在冲突可能,进而引发权利纠纷。为了解决不同开源许可证的兼容性问题,美国国防部研究所David Wheeler博士提出了快速确认兼容的方法^[5](参见图2)。

根据图2可知:

(1)方框之间用箭头连接,称箭头起点为方框A,箭头终点为方框B。 $A \rightarrow B$ 代表许可证A是许可证B兼容的,即结合的作品可以以许可证B发布。

(2)判断程序是不是可以被组合起来,只要从它们各自的许可证开始,顺着箭头方向看能不能找到一条路径,若能形成一条路径则可以组合,组合后的结果是适用方框B中的许可证(即以B许可证发布)。例如,遵循Apache 2.0许可证的程序和遵循GPL v2以上版本的许可证都可以到达GPLv3 or GPLv3+,所以它们可以在GPLv3 or GPLv3+许可证下组合使用。

(3)“ \rightarrow ”方向是不可逆。例如:Apache 2.0不是GPL v2兼容的(即结合的作品不能以GPL v2发布);Apache 2.0许可证规定,加入到该软件的一切代码,都必须授权其代码中包含的任何专利的使用权。这意味着,该专利不会对Apache软件基金会和Apache使用者收取费用。而对于专利

授权,GPL v2中并没有明示。Apache 2.0是GPLv3是兼容的:GPL v3包含有明确的专利认可条款,意味着任何向GPL项目捐赠代码的实体同意将软件包含的专利永久性免费许可给该项目。

4 开源软件的选择建议

根据上述不同开源许可证的传染性和兼容性特点,可以将开源许可证分为三类:强(GPL/LGPL/AGPL)、弱(MPL/EPL)、宽容型(MIT/BSD/Apache)。按照不同的角色分类,给出以下建议:

4.1 软件使用者的选择建议

软件使用者是指直接运行软件的企业或个人,根据开源软件免费使用的特点,一般使用者可以选择使用任何类型许可证的开源软件,但是需要注意的是:

(1)如果选择使用“强”许可证的开源软件,可能导致软件整体负有开源义务,即:使用者对开源软件的任何优化、修改部分均需再开源。

(2)如果使用开源软件向他人提供在线服务(例如:舆情通、商情通),则不能采用遵循AGPL的开源软件。因为在此种情况下,使用人仍负有开源的义务。

4.2 技术服务提供者的选择建议

技术服务提供者是指为他人提供基于开源软件的部署、升级、二次开发等技术服务的个人或企业。理论上,可以选择任何类型的开源软件,但从考虑自身技术优势保护的角度,尽量要避免为客户修改或新开发的代码被迫开源。因此,在选择上,也应尽量避开选择“强”、“弱”许可证。

值得注意的是,任何企业一旦选择了“强”、“弱”许可证,也就意味着针对该开源软件的商业模式,只能选择提供技术服务的方式。因为,由于遵循“强”、“弱”许可证的开源软件,修改或新增部分被传染后,贡献者被迫放弃了许可使用权利,软件产品自然就失去其经济价值。

4.3 软件产品提供商的选择建议

软件产品提供商是指通过开发软件产品(闭源软件),并以售卖产品许可使用权获得对价的企业或个人。如果要选择基于开源软件开发一款软件产品,那么“宽容型”开源软件是最佳选择。如果,在软件产品中不可避免的要使用到“强”、“弱”型开源软件,则需十分谨慎,并要从技术层面进行隔离,避免被引用的开源软件传染。

此外,企业在选择开源软件时,还要综合考虑企业自身技术能力,如:学习成本,学习曲线是否陡峭,能否在短时间内掌握并运用,是否影响项目进度。以及考虑开源社区活跃度(是否有丰富的相关技术问题的讨论和分享,遇到困难时可否寻求帮助)、更新情况(版本更新是否频繁,是否有严重的安全问题或缺陷)、应用广泛性(有哪些公司在使用该技术,是否有案例分享)、生态圈(技术生态圈是否成熟)等因素,最终确定自己的技术路线和商业模式。

参考文献

- [1] 糜凯,史尚元.开源软件的知识产权研究[J].现代情报,2009,29(09):4-6.
- [2] 张平,马骁.共享智慧:开源软件知识产权问题解析[M].北京:北京大学出版社,2005:87.
- [3] 蔡俊杰.开源软件之道[M].北京:电子工业出版社,2010:42-52.
- [4] GNU Operating System.Frequently Asked Questions about version 2 of the GNU GPL[EB/OL].<http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#WhatIsCompatible>,2018-7-4.
- [5] David A.Wheeler: The Free-Libre / Open Souse Software (FLOSS) License Slide[DB/OL]. <https://hardenedlinux.github.io/2016/10/21/best-practices-criteria-for-floss-part3.html>.

Legal Risk Analysis and Prevention of Open Source Software

LI Bin, DU Tao, QIAN Ling

(Zhongchang (Suzhou) Software Technology Co., Ltd., Suzhou Jiangsu 216153)

Abstract: With the development of "open source software" ecological chain, the number and influence of open source software are increasing. Because of the low cost and fast upgrade of open source software, more and more enterprises choose to use open source software. But "there is no free lunch in the world," and under the constraints of various open source licenses, users of open source software are not free and free as they imagine. The improper use of open source software may not only cause direct economic losses and negative public opinion to users, but also lead to the loss of users' technology. Based on the legal analysis of open source software and open source license, combined with the application practice of open source software in a product, this paper summarizes the usage rules of open source software, and puts forward some suggestions for the compliance improvement of the product.

Key words: open source software; license; compatibility