

开源许可证对比研究

赵亮*, 高超, 苏晓艳

军事科学院系统工程研究院, 北京

收稿日期: 2021年11月19日; 录用日期: 2021年12月16日; 发布日期: 2021年12月23日

摘要

开源许可证是开源软件的灵魂, 但是鲜有开源参与者和使用者能够充分阅读开源许可证的原文全文, 理解开源许可证的要求约束, 使得开源活动中引入了各种潜藏的风险。通过介绍开源许可证的相关定义, 分析开源许可证的基本性质, 对比分析MIT、BSD、Apache和GPL等几种主要的英文开源许可证和以中文为书写语言的木兰许可证的许可条款、演变历史, 详细阐述了这几种开源许可证的发展轨迹、关键条款、授权范围和许可条件, 对比分析了开源许可证之间以及开源许可证与商业许可证的差异, 明确了各开源许可证的主要特点和在开源活动中的基本遵循, 以帮助开源软件的参与者和使用者充分理解开源精神, 提高开源活动中的行为符合性, 促进开源软件在中国的发展。

关键词

开源软件, 许可证, 著作权, 著佐权, MIT, BSD, Apache, GPL

A Comparative Study on Open Source Licenses

Liang Zhao*, Chao Gao, Xiaoyan Sun

Department of System Engineering, Academy of Military Sciences PLA, Beijing

Received: Nov. 19th, 2021; accepted: Dec. 16th, 2021; published: Dec. 23rd, 2021

Abstract

Open source license is the soul of open source software, but few open source participants and users can fully read the full text of the open source license and understand the requirements and constraints of the open source license, which leads to various potential risks in the open source activities. This paper introduces the definition of open source license, analyzes its basic nature,

*通讯作者。

compares and analyzes the license terms and evolution history of several main English open source licenses such as MIT, BSD, Apache and GPL, and Mulan license written in Chinese. It also elaborates the development track, key terms, authorization scope and license conditions of these open source licenses, and makes clear that the main characteristics of each open source license and the basic compliance in the open source activities, makes comparison among open source licenses and between open source license and business license, which can help the participants and users of open source software fully understand the open source spirit, improve the behavior compliance in the open source activities, and promote the development of open source software in China.

Keywords

Open Source Software, License, Copyright, Copyleft, MIT, BSD, Apache, GPL

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

开源软件经过 30 年左右的迅猛发展, 改变了软件产业格局、构建了开源软件生态、形成了开源开发模式, 并催生了众包经济形态[1]。开源软件已成为互联网时代和移动互联网时代的信息基础设施。IDC 和 Gartner 的调研报告都显示, 有 99% 的企业在软件开发中使用开源软件[2]。RedHat 调查报告显示 95% 的 IT 领导者认为开源软件对于企业未来的发展和创新极为重要[3]。对我国国防科技和军事科研中开源应用情况调研显示, 开源软件在该领域的应用已非常普遍[4]。开源软件在各领域的应用已成不可阻挡之势。

开源许可证是开源软件应用中的基本遵循, 但是由于用户习惯、书写语言或者知识产权意识淡薄, 众多的开发参与者对开源许可证的内容缺乏必要的学习和理解, 造成开源应用中的风险[5], 由于违反开源许可造成的诉讼损失有的高达几亿美元。近年, 我国也产生了开源软件的多个判例[6]。如何合理使用开源软件, 降低开源应用中的风险, 是开源软件应用中必须注意的关键问题[7]。如何合理、科学地使用开源软件来进行二次创新, 是当今软件研发必须掌握的基本知识, 更是国防科技领域软件研发和开源治理的重要内容。

2. 开源许可证的基本特征

2.1. 基本定义

定义 1: 著作权(Copyright): 亦称为版权。作者、著作权人依法对作品享有的权利, 包括人身权和财产权。前者包括发表权、署名权、修改权和保护作品完整权; 后者包括复制权、发行权、信息网络传播权、摄制权、改编权、翻译权和汇编权。(《世界版权公约》是著作权国际保护的重要规则, 中国于 1992 年加入) [8]。

定义 2: 软件著作权(Software Copyright): 指软件的开发者或者其他权利人依据有关著作权法律的规定, 对于软件作品所享有的各项专有权利。依据《计算机软件保护条例》, 包括发表权、开发者身份权、使用权和获得报酬权等[8]。

定义 3: 著佐权(Copyleft): 与 Copyright 相对, 是自由软件运动提出的概念, 是利用现有著作权体制来保护软件的所有用户和二次开发者的一种授权方式, 要求使用者修改后的衍生作品必须要以同等的授

权方式释出以回馈社会[9]。

定义 4: 许可证(License): 可撤销的书面(正式)或默示协议, 表明在特定时期和特定的条件下, 授权方或许可方不主张他或她自身的权利, 以允许其他方(受许可方)从事某些正常情况下被禁止的活动。知识产权方面的许可意味着即使被许可的对象(艺术, 设计, 专利等)被受许可方复制、出售或使用, 许可方也不会行使所有权的法律保护[10]。

定义 5: 自由软件基金会(Free Software Foundation, FSF): 自由软件运动的发起机构和倡导者, 致力于推广自由软件、促进计算机用户自由的美国民间非盈利组织。成立于 1985 年 10 月, 总部位于美国马萨诸塞州波士顿[9]。

定义 6: 开源软件促进会(Open Source Software Initiative, OSI): 开源软件的标准机构, 维护开源定义, 致力于倡导、推动和普及开源软件。1998 年 2 月从 FSF 中独立, 总部位于美国加利福尼亚州帕洛阿托(Palo Alto) [11]。

计算机软件随着计算机硬件的产生而产生, 1957 年美国管理学家 John Wilder Tukey 首先提出和使用了“软件”(Software)一词。软件概念的出现标志着计算机软件成为与计算机硬件相对应的独立一部分。1969 年美国司法部责令 IBM 公司停止把软件与其硬件捆绑销售, 从此软件在法律意义上成为独立于硬件的产品。1968 年, Informatics 公司制定了软件商业授权协议, 创造了软件独特的盈利模式。

软件著作权是传统著作权保护在软件领域的延伸, 重点在于保护著作权人的权利。在软件著作权保护下, 一方面, 软件开发者的利益得到了彰显, 形成了商用软件盈利模式, 催生了微软、甲骨文等软件商业巨头, 集聚了大量的社会财富; 另一方面, 著作权也限制了软件产品的自由流通和使用, 限制了思想的自由交互[12]。随着互联网的兴起和移动互联网的普及, 这种以独占知识为基础的系统越来越表现出不适合他们自己所创造的世界[12]。

1983 年, Richard M. Stallman 发起了自由软件运动, 旨在保护用户对软件使用、修改和分发的自由。自由软件运动产生了大量优秀的自由软件, 为防止商业软件窃取了自由软件的开发成果, Richard M. Stallman 创造性地提出了著佐权概念。GPL 许可证就是著佐权的完整表达[9]。

与著作权保护将权利授予作品相关的人或机构不同, 著佐权许可证将权利保留在软件产品(或项目)本身, 要求所有的参与者和贡献者保持软件项目的自由分发和共享状态。类比而言, 著作权将软件的各项权利作为“物权”授予开发者, 而著佐权则将软件的各项权利作为“人权”保留在项目本身。开源软件许可证更加商业友好, 一方面要求保证开源软件自由传播的延续性[13], 另一方面, 通过允许商业化的参与, 给软件行业创造了前所未有的自由空间, 带来了软件行业的繁荣发展[14]。

2.2. 基本性质

开源许可证本质是一个权利合同。开源许可证虽然不同于版权登记, 但本质上也是一种法律性质的合同, 美国《统一计算机信息交易法(Uniform Computer and Information Transactions, ACT)》和《中华人民共和国民法典》都认定开源许可证是一个权利合同。开源软件通过开源许可证, 强制任何使用和修改该软件的人承认发起人和所有参与者的贡献, 同时, 任何人都拥有自由复制、修改和使用这些源代码的权利, 并且不得设置针对任何人或团体领域的限制、不得限制开源软件的商业使用等[15]。

开源许可证是开源软件的灵魂, 开源代码是开源软件的肉身。贡献者、开发者和使用者在参与开源项目研发、使用开源软件的过程中, 更多是被“肉身”吸引, 而很少考察其“灵魂”。一般开发者和用户鲜有人仔细浏览许可证的内容, 更少有人能把许可证内容完整看过一遍[5]。原因在于, 一方面, 大部分许可证都是用英语写的, 对于中国的开源参与者, 非母语语境使得他们通读或详细了解许可证的机会较小; 另一方面, 知识产权观念淡薄。虽然, 大多数情况下, 许可证约束的条件或情境不会发生, 涉事

者只要关注开源软件的功用即可，所谓“暗合天道，无问西东”。但是，时或报道出来的由于开源软件使用造成的知识产权纠纷和巨额诉讼案件还是在警醒参与者，“开源有风险，使用需谨慎”。尤其在在我国防军事领域或其它强监管领域，由于制度性限制，使得实际应用中很难严格遵循许可证要求，对此带来的风险要慎之又慎。

开源许可证种类繁多，许可类型多样。数据显示，目前在用的许可证超过 3000 个，SPDX 官网发布的许可证有 444 个[16]，自由软件基金会(FSF)认证的有 106 个，开源软件促进会(OSI)认证的有 126 个，共同认证的有 61 个。这些许可证，不仅内容不同，文风也有很大差别。面对如此繁多的许可证，难免会陷入选择综合症。根据对权利约束和传染性情况的不同，中国信息通信研究院将开源许可分为开放型许可、弱传染性许可、传染性许可和强传染性许可四类[17]。不同类型的许可证对权利主张要求和方式不同。其中细微的差别会对使用者和参与者产生哪些方面的影响，需要根据具体的场景进行具体的分析。

3. 主要开源许可证分析

3.1. 开源许可证选择

根据 GitHub 和 Gitee 代码托管网站的统计结果，采用比例排名前十的许可证如表 1 所示。基于两个网站的统计结果，选择 MIT、BSD、Apache 和 GPL 许可证作为重点分析对象，并结合中国开源研究的成果，分析木兰许可证相关条款。

Table 1. Adoption Proportion of open source license

表 1. 开源许可证采用比例

#	GitHub 网站数据		Gitee 网站数据	
	许可证	比例	许可证	比例
1	MIT	44.69%	Apache-2.0	40.9%
2	Other	15.68%	MIT	34.8%
3	GPLv2	12.96%	GPLv2	6.8%
4	Apache	11.19%	GPLv3	6.4%
5	GPLv3	8.88%	BSD 3-clause	2.5%
6	BSD 3-clause	4.53%	LGPLv3	1.80%
7	Unlicense	1.87%	AGPLv3	1.05%
8	BSD 2-clause	1.70%	BSD 2-clause	0.70%
9	LGPLv3	1.30%	LGPLv2.1	0.80%
10	AGPLv3	1.05%	AFL-3.0	0.5%

注：1) <https://www.github.com/>，Github 为全球最大的开源代码托管网站；2) <http://www.gitee.com.cn/>，Gitee 为中国最大的开源代码托管网站。

3.2. MIT 许可证 - 历久弥新的许可证

MIT 许可证是最古老的软件许可证之一，从最初的发布至今天，MIT 许可证内容并没有太大的改变，但是该许可证历久弥新，依然保持着旺盛的力量[18]。在 GITHUB 的调查中，基于该许可证发布的软件是比例最大的。从另一个侧面说明，最简单的东西也是最强大最有生命力的。

3.2.1. 关于 MIT 许可证的名称

FSF 网站上将 MIT 许可证称为 Expat License 许可证，OSI 网站使用 MIT 这个称呼。FSF 认为使用“MIT 许可证”这个称呼具有一定的误导性，原因在于：一方面，MIT (麻省理工学院)为其发布的软件使用了

很多的许可证，本许可证只是其中之一；另一方面，很多人将 X11 (X windows 系统的第 11 个版本)许可证称为 MIT 许可证，或许折中的做法是叫做 MIT X 许可证。X11 第一个版本诞生的时间是 1987 年 9 月 25 日。时至今日仍然被所有类 Unix 系统广泛使用，如企业版 Linux 发行版、BSD 和 Solaris 版本。

3.2.2. MIT 许可证授权内容

从授权内容看，MIT 许可证是一个宽松的、非著佐权型的许可证，授予任何人无限制地处理软件的权利，包括但不限于使用、复制、修改、合并、发布、分发，再许可和/或出售软件，该许可不限于授予任何特定的知识产权。

本许可证提到了免责声明，但是并没有提到专利的问题。因此 FSF 认为该许可证存户利用专利漏洞提起诉讼的可能性。但是也有人认为 MIT 是一个明示的许可证(Express license)，“没有明确提及专利权并不意味着没有授予专利权的明示许可，也并不意味着仅限于默示授予专利权” [18]。

MIT 许可证的不主张版权要求，允许开发者按照他们自己的需要自由地修改软件。或许正是由于这种灵活性，因此被认为是最灵活的许可证并被广泛使用。

3.3. BSD 许可证 - 日渐精简的许可证

BSD 许可证是最早的许可证，最初发布在 1982 年，属于加利福尼亚大学董事会，由加州大学伯克利分校具体管理。

3.3.1. BSD 许可证的演化

原版 BSD 许可证(亦称为旧版 BSD 许可证、4 条款 BSD 许可证)包含四条内容，分别描述了源码分发、目标代码分发、必须要展示的内容(广告条款)和不能用版权所有者信息作衍生品推广等四方面的要求。

根据广告条款约定，软件运行时要显示所有软件贡献者的信息。随着软件贡献者越来越多，该条款给软件带来的负担就越来越大。因此，到 1999 年 7 月，BSD 删除了广告条款的要求，变成了新版 BSD 许可证(亦称为修改版 BSD 许可证、3 条款 BSD 许可证)，并正式废止了 4 条款 BSD 许可证。

再后来，BSD 又删除了关于产品推广的约定条款，变成了 2 条款 BSD 许可证。

总体而言，随着时间的发展，BSD 许可证越来越瘦身，限制性条款越来越少，许可要求似乎也越来越灵活。但是对许可证而言，也并非限制越少越好，因为很多限制性的条款包含了对开发者的保护。实践表明，3 条款的 BSD 许可证是 BSD 许可证中应用是最多的。

此外开源界还有许多 BSD 许可证的变形。如 1 条款 BSD 许可证，内容上它只保留了原 BSD 许可证中的第一条要求，但该许可证产权属于伯克利软件设计公司，与上述 BSD 许可证不属于同一属主，只是经 BSD 许可证管理者的同意，借用了这个名字。

3.3.2. BSD 许可证的授权内容

新版 BSD 许可证(亦称为修改版 BSD 许可证、3 条款 BSD 许可证)有三方面的约定。

- 1) 版权规定：保留完整的版权声明
- 2) 源码分发和目标代码分发时应遵循的条件
- 3) 不得将版权所有人或贡献者姓名用于衍生品的宣传和推广

与 MIT 许可证类似，BSD 许可证中无专利条款。贡献者可以在软件中布局专利，并可通过专利授权收费；另一方面，由于程序接收者没有获得版权所有者、贡献者关于专利的授权，存在侵权风险。

3.4. Apache 许可证 - 包容并蓄的许可证

Apache 许可证是 Apache 基金会基于 BSD 和 MIT 许可证要求，结合社会发展和技术条件变化，增加

一些重要条款而形成。最初仅为 Apache HTTP 服务器项目使用，后来逐渐应用到 Apache 基金会管理的更多开源项目器，由此被推广，成为目前最受欢迎的许可证。

3.4.1. Apache 许可证的授权内容

Apache 许可证属于宽容型许可证，主要表现在：

- 1) 版权：它授予用户永久性的、全球性的、非排他性的、免费的、不可撤销的版权许可证。
- 2) 商标：限制商标使用。许可证并未授予用户使用许可证颁发的商号、商标、服务标记或产品名称，除非是为合理和惯常的方式描述作品来源和复制通知文件有关内容所需。
- 3) 再分发：允许用户以不同的许可再分发修改过的作品(衍生作品)。

3.4.2. Apache 许可证的专利反制

与 MIT 和 BSD 许可证不同的是，Apache 许可证中定义了明确的专利条款，要求：

每个贡献者授予用户永久性的、非排他性的、免费的、无许可费的、不可撤销的专利许可。

同时，Apache 许可证还定义了“专利报复”条款即“如果用户就作品或作品中的所涉及的贡献起诉，则根据本许可证，授予用户针对作品的任何专利许可证在提起上诉诉讼之日起终止”。通过这两方面的定义，Apache 许可证既保护了用户自由，也保护了贡献者权利，从而更好地保证了开源项目的推广应用。

3.5. GPL 许可证解读 - 著佐权许可证的旗帜

GNU 通用公共许可证(GPL)是 FSF 的标志性许可证，源自 FSF 对软件自由的坚持和追求。FSF 认为软件自由包括四个方面：使用自由、修改自由、分发自由和分发修改版的自由，用户行使以上自由权利的基础是软件源码可获得[5]。基于这一基本观点，GPL 定义了一个允许持续共享已发布作品的许可证框架，既授予用户自由，又保护用户自由。GPL 体现着自由软件的精神，代表了 Copyleft (著佐权)的具体实现和表达。

3.5.1. GPL 许可证的授权内容

GPL V1.0 诞生于 1989 年 2 月，是自由软件运动的创始人 R. Stallman 为 GNU 项目设计的第一个许可证。GPL V1.0 共有 10 条约定，包括：

- 1) 拷贝和分发软件的自由权利：任何人在保持许可证完整的条件下都可以自由的拷贝和分发基于 GPL 许可的软件。
- 2) 修改和分发修订软件的权利：标识修改的内容和时间，向第三方免费传递许可证，有交互界面时如何传递许可证，与其它独立程序聚合时不需传递许可证。
- 3) 分发目标代码的权利和约定：跟目标代码一起去随附源代码、给予提供源代码的三年承诺、提供可以从哪里获取源代码的信息
- 4) 违反许可证的行为约束：违反许可证的行为将终止许可的权利
- 5) 许可证的默许遵循：使用、修改和分发软件的行为表示默许遵循许可证
- 6) 许可证的自动传递：不得向后续接收者强加限制
- 7) 许可证的引用形式：如何在程序中使用许可证
- 8) 许可证的组合：经过 FSF 商讨，可以进行许可证的组合发布
- 9-10) 免责条款

这些条款形成了 GPL 的基本框架，后续版本在此基础上，根据技术的进步和社会环境的变化，逐渐进行了完善和细化。GPL2.0 颁布于 1991 年 6 月，作了如下修改：

- 1) 关于术语的严格定义：增加了更多的人称；

- 2) 细化了软件修订的各种情况,对软件修订、软件聚合等进行了区分,明确了各种条件下如何保持许可证的完整性和传递用户权利;
- 3) 细化了分发目标代码时,提供源代码的几种方式;
- 4) 增加了关于专利的许可项:既防止贡献者利用专利起诉用户,也限制第三方主张软件中的专利权;
- 5) 如果许可证要求与所在国法律有冲突时应如何保护权利。

GPL3.0 颁布于 2007 年 6 月,主要修订包括:

- 1) 明确了专利许可:任何贡献者必须依据其必要专利声明,给予程序使用者非排他、全球性的免费制造、使用、销售、许诺销售、进口或运行、修改、传播贡献者版本内容的权利。但是,与 Apache 许可证不同,GPL 中没有专利反诉条款。

- 2) 无明确的商标规定,也无商标使用授权。

- 3) 明确了违约责任。如果没有按照许可证传播或修改覆盖程序,权利将自动终止。但对违约者态度比较温和,只要求其在指定时间内(30 天)修正违反行为,就可以继续享受授权。

- 4) 对 DRM (数字版权管理)的限制。

绝大多数对 GPLv3 有争议的条款都涉及到对数字版权的管理。新版本的 GPL 并没有要求禁用 DRM,但是通过声明和允许家用设备的用户在产品中执行自己修改的软件版本这两种途径对它进行了限制,从而阻止使用 DRM 时对互操作性的限制,保证了用户的自由。

3.5.2. GPL 许可证的引用模式

从版本看:GPL 目前有 3 个版本,但是从引用方式上,每个版本又有 2 种引用方式,即 only 方式(现在时)和 later 方式(未来时)。这两种引用方式引用的文本内容完全一致,只是在项目引用时许可证的头文件表述有区别,only 方式的引用说明是,“This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3.”表明该项目仅基于当前许可证版本发布。

而 later 方式的引用说明是“ This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.”说明后续的许可证修订版本也可以适用于本项目。

产生这 2 种许可证引用方式的原因在于 FSF 认为允许软件的许可证升级对使用不同版本 GPL 发布的程序保持兼容性非常重要。Later 方式这种描述方式是 FSF 构造的允许用户将程序许可证升级到后来版本的许可框架。当诸如 GitHub 这样的网站邀请开发者选择按照“GPL 3”或者“GPL 2”等许可证来发布时,并没有提出许可证将来版本的问题,这就导致数以千计的开发者没有理清他们代码的许可证。如果请这些用户在“only”和“later”之间选择,那么其代码许可证问题就会比较清晰。这也避免了许可证未来的不兼容性。FSF 鼓励开发者选择“later”这样的许可方式,给用户根据意愿按照 GPL V1.0 版本、GPL V2.0 版本或 GPL V3.0 版本,甚至 GPL V4.0 版本(如果将来发布的话)来使用程序的自由。这样,FSF 通过“only”和“later”这两种引用方式,从而让开源代码的托管社区明白这些不同,并且鼓励开发者清楚地表述他们的决定。

3.5.3. GPL 许可证协议簇

GPL 许可证是一个协议簇,还包括 LGPL (Lesser GPL)和 AGPL (Affero GPL)。这两款许可证的内容与 GPL 的内容基本一致,但是分别针对服务器软件和函数库软件提出了相应的限定条件,其中:

LGPL 是针对函数库的许可,它允许私有软件使用该函数库,允许私有软件和开源软件并存。

AGPL 是针对服务器软件(或者网络软件)的许可证, 避免有人用应用服务提供商的方式来逃避 GPL 责任。如果一个以 AGPL 授权的软件被运行在向公众提供服务的服务器上, 那么对该软件的任何修改也必须要以源代码的形式对外发布。

由于有以上的不同, 通常将 GPL 和 AGPL 许可认为是强传染性的许可证, 而 LGPL 为弱传染性或者宽松的许可证。

从版本号看, LGPL 的第一个版本是 V2.0, 于 1991 年在 GPL2.0 的基础上颁布, 1999 年升级为 V2.1, 2007 年 6 月与 GPL 许可证同步升级为 V3.0。AGPL 的第一个版本就是 V3.0, 表明其与 GPL 许可证版本的伴随性。

3.6. 木兰许可证——第一款为 OSI 认证的中文许可证

3.6.1. 中文的许可效力

木兰许可证 2020 年 2 月 14 日通过 OSI 认证, 是第一款通过 OSI 组织认证的中文许可证。该许可证是国家重点研发计划“云计算和大数据开源社区生态系统”的子任务, 由北京大学牵头, 依托全国信标委云计算标准工作组和中国开源云联盟, 联合国内开源生态圈产学研各界优势团队、开源社区以及拥有丰富知识产权相关经验的众多律师, 共同研制而成[19]。

木兰许可证以中文语言的表述为主。许可证中明确说明“本许可证”以中英文双语表述, 中英文版本具有同等法律效力。如果中英文版本存在任何冲突不一致, 以中文版为准。在此之前, 所有的许可证都是用英语表达的, GPL 许可证明明确表明“只有 GNU 通用公共许可证英文原版才具有法律效力”[9]。因此, 拥有一个中文语言的许可证对于中国的开源界而言是一个巨大的进步。

与其它许可证由民间组织发布和维护不同的是, 木兰许可证是由中国官方组织设计和颁布的, 既表明政府对开源的支持态度, 同时对中国开源界具有更好的指导作用。

3.6.2. 木兰许可证的授权内容

从内容看, 木兰许可证授予版权、专利许可, 无商标许可, 并明确了分发限制, 给出了免责声明与责任限制, 在专利部分明确了专利反制权利。

木兰许可证第一版颁布于 2019 年 8 月, 第二版颁布于 2020 年 1 月, 两个版本之间并没有本质的或者实质性的差别, 通过修改, 在文字上更加简洁, 限定对象上更加明确:

1) 调整了定义出现的顺序, 将“贡献”从原来的第五个提升到第二个, 以便于解释其后的“贡献者”含义, 表述更符合逻辑性。

2) 略微修改了“关联实体”的定义, 从原来的“对本许可证下的一方而言”, 修改为“对本许可证下的行为方面而言”, 使得对象的限定更加明确。

3) 修改了“授予专利许可”的条款描述。从“不包括仅因您或他人修改‘贡献’或其他结合而将必然会侵犯到的专利权利要求”, 修正为“不包括对‘贡献’的修改或包含‘贡献’的其他结合”。

4. 开源许可证比较

4.1. 开源许可证的共同点

横向对比 MIT、BSD、Apache、GPL 以及木兰许可证等常见许可证, 这些许可证的共同之处主要有以下几点:

1) 授予用户使用、复制、修改、再发行软件的权利。开源软件区别于商用软件的特征就是将商业软件保留于开发者的权利全部授予了用户和潜在的贡献者。

2) 要求保持公开源码的义务。用户修改软件最佳的条件是能够获得软件的源代码。为了促进开源软件的传播, 保证开源软件的持续发展, 开源软件许可证都会规定各级发行人有公开源码的义务。

3) 修改情况说明。开源软件许可证要求发行者将著作权许可申明、软件功能说明、源代码修改情况、修改日期等信息以适当方式在发布的软件中给予标注。

4) 普遍存在“不担保”条款。“不担保”是指, 对开源软件中存在缺陷、漏洞, 而给用户造成损失或产生软件知识产权纠纷, 开源软件的发行者并不承担法律责任。而且开源软件许可证“不担保”条款涉及的范围很广, 不论造成的损失是一般还是严重, 直接还是间接, 也不论是违约责任还是侵权责任, 发行者都不承担责任。

4.2. 开源许可证的不同点

开源许可证的不同主要在于不同许可证对于开源软件的商业化使用、分发、修改、专利、闭源发行等方面的继生权利要求不同, Choose an open source license|Choose a License 网站详细比较了各种不同许可证在以上几个方面的不同。根据常见许可证的传染性 and 对于专利许可的规定, 可以将其总结如图 1 所示:

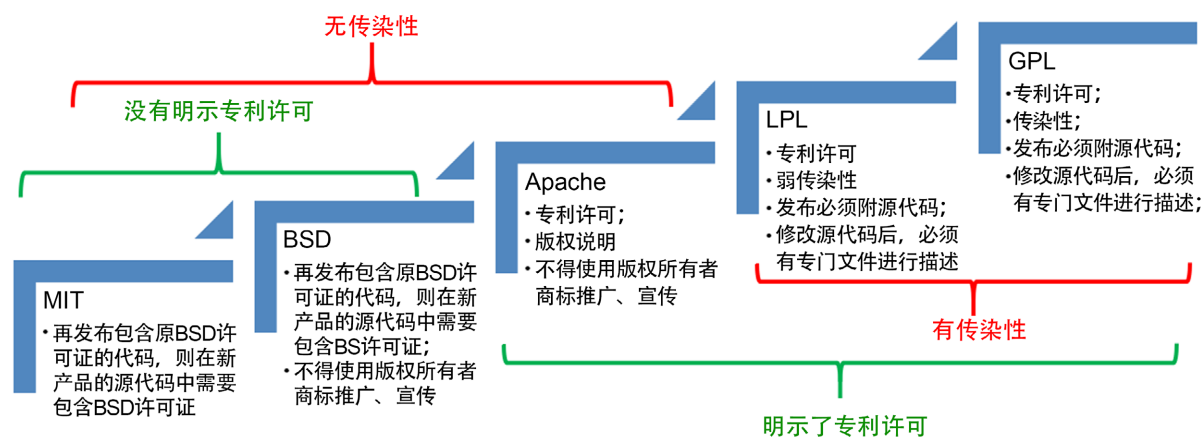


Figure 1. Comparison of open source right

图 1. 开源许可证权利特征比较

1) 从传染性看: MIT、BSD、Apache 和木兰等许可证属于宽容型的许可证, 允许项目代码被商业软件和闭源软件使用, 且允许演绎作品使用其它许可证发行; 而 GPL (LGPL) 许可证则有很强的传染性, 要求相关的代码继续开源, 并要求演绎作品保持相同的许可证发行。

2) 从专利保护角度: MIT 和 BSD 许可证没有明示的专利许可; 而 Apache、GPL (LGPL) 和木兰许可证中都有明示的专利许可, 保护项目中的代码特征。

3) 从兼容性看: 这些许可证都是与 GPL 许可证兼容的许可证。即基于 MIT、BSD、Apache2.0 和木兰许可证发布的软件都可以与基于 GPL 许可证发布的软件组合成一个大的软件程序发布。

4.3. 开源许可证与商业许可证比较

与商业许可证相比, 开源软件也是有版权的。其权利人是开发此软件的所有贡献者或由特定国家的法律确定的权利人[20]。只是因为采用了开源许可证, 权利人放弃了独享复制、修改、分发的权利, 而是要求保证将这些权利顺次传递下去, 形成了开源软件独特的供应链[21] [22] [23]。各种软件许可证仍是一种版权保护规则, 只是传统的版权保护规则的目的、性质发生了改变, 并做了一些与传统版权法相反

规定。开源许可证与商业许可证的比较如表 2 所示。

Table 2. Comparison of Software Licenses

表 2. 开源许可证与商业许可证比较

#	开源许可证	商业许可证
1	授予接受者永久的软件使用、复制、修改和再发布的权利	接受者仅获得软件的使用权(或有期限或使用范围限制)
2	要求接受者将获得的权利持续传递下去	接受者不得传递获得的权利
3	许可的甲方根据贡献者的不断增加而不断变化的	许可证的甲方只能是软件的原始发行者
4	免除软件发布人的瑕疵担保责任	发布人依许可协议承担有限瑕疵担保责任
5	采用以行为做出承诺的订立方式	采用启封许可证和点击许可证方式订立
6	以电子数据为表现方式的许可证	启封许可证一般情况有纸质合同表示, 点击许可证以电子数据表示

5. 结论及应用建议

本文介绍了 MIT、BSD、Apache、GPL 和木兰等开源许可证的演化历史, 详细分析了开源许可证的许可条款, 并对比分析了开源许可证以及开源许可证与商业许可证的区别, 为开源项目选择许可证、提高开源项目的许可证符合性提供指导。

从分类看, MIT、BSD、Apache 和木兰均是宽松的许可证, 允许对作品自由的使用、修改、发布和再发布, 在版权、专利等使用上相对宽松, 允许与其他许可证混合使用, 甚至允许演绎作品闭源使用。GPL 是严格的著佐权许可证, 充分保护用户的自由, 并严格保护软件代码及其衍生作品的自由, 不得与其他许可证混合使用, 以降低代码的自由度。严格的著佐权许可证树立了自由软件的标杆, 开创和保证了软件自由共享的新模式; 而商业友好的许可证则极大地推动了开源软件的普及和拓展, 促成了开源软件在世界范围内的星火燎原之势, 成为移动互联时代的基础设施[1] [7]。

从国内应用看, 大数据、云计算、人工智能和物联网领域的大量基础软件和应用软件都有开源的影子, 开源软件成为我软件领域与世界同步的重要渠道, 但同时, 必须认识到, 我国国防科技领域和一些强监管领域, 由于制度上的限制, 很难严格遵循某些许可证的限制要求。因此, 在开源软件使用和开源治理上仍有一系列的重要问题需要我们认真对待, 只有不断努力, 才能在世界科技格局变化中, 占得先机, 赢得主动, 稳定前行。

参考文献

- [1] 梅宏. 我国开源软件技术发展策略建议(咨询报告) [R]. 2016.
- [2] Gartner Hyper Cycle for Open Source Software. July, 2016.
- [3] 企业开源现状[EB/OL].
<https://www.redhat.com/cms/managed-files/rh-enterprise-open-source-report-detail-f21756-202002-a4-zh.pdf>
- [4] 赵亮. 混源软件项目中开源组件的影响分析[J]. 计算机科学, 2020, 11(11A): 3075-3078.
- [5] Almeida, D.A., Murphy, G.C., Wilson, G., *et al.* (2017) Do Software Developers Understand Open Source Licenses? 2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC), Buenos Aires, 22-23 May 2017, 1-11. <https://doi.org/10.1109/ICPC.2017.7>
- [6] <https://www.court.gov.cn>
- [7] 新思科技. 开源安全和风险分析报告(2021) [R].

-
- [8] 著作权[EB/OL].
<https://dict.baidu.com/s?wd=%E8%91%97%E4%BD%9C%E6%9D%83&device=pc&from=home>
- [9] <https://www.fsf.org>
- [10] <http://www.businessdictionary.com/definition/license.html>
- [11] News|Open Source Initiative.
- [12] 蒲实, 陈赛. 大学的精神[M]. 北京: 中信出版社, 2017: 164-165.
- [13] Schoettle, H. (2019) Open Source License Compliance—Why and How? *Computer*, **52**, 63-67.
- [14] Ballhausen, M. (2019) Free and Open Source Software Licenses Explained. *Computer*, **52**, 82-86.
<https://doi.org/10.1109/MC.2019.2915690>
- [15] Wheeler, D.A. (2007) The Free-Libre/Open Source Software (FLOSS) License Slide.
<http://www.dwheeler.com/essays/floss-license-slide.pdf>
<https://doi.org/10.1109/MC.2019.2907766>
- [16] The Software Package Data Exchange® (SPDX®). <https://www.spdx.org>
- [17] 云计算开源产业联盟. 开源产业白皮书(2019) [R].
- [18] Peterson, S.K. (2018) Why So Little Love for the Patent Grant in the MIT License.
- [19] license.coscl.or.cn.
- [20] Kapitsaki, G. and Charalambous, G. (2019) Modeling and Recommending Open Source Licenses with findOSSLicense. *IEEE Transactions on Software Engineering*, **47**, 919-935.
- [21] Coughlan, S. (2020) Standardizing Open Source License Compliance with OpenChain. *Computer*, **53**, 70-74.
- [22] 孙福洲, 钱瑾, 杨静, 等. 云服务时代下的开源发展[J]. 电信科学, 2020, 36(11): 156.
- [23] 梁冠宇, 武延军, 吴敬征, 赵琛. 面向操作系统可靠性保障的开源软件供应链[J]. 软件学报, 2020, 31(10): 3056-3073.