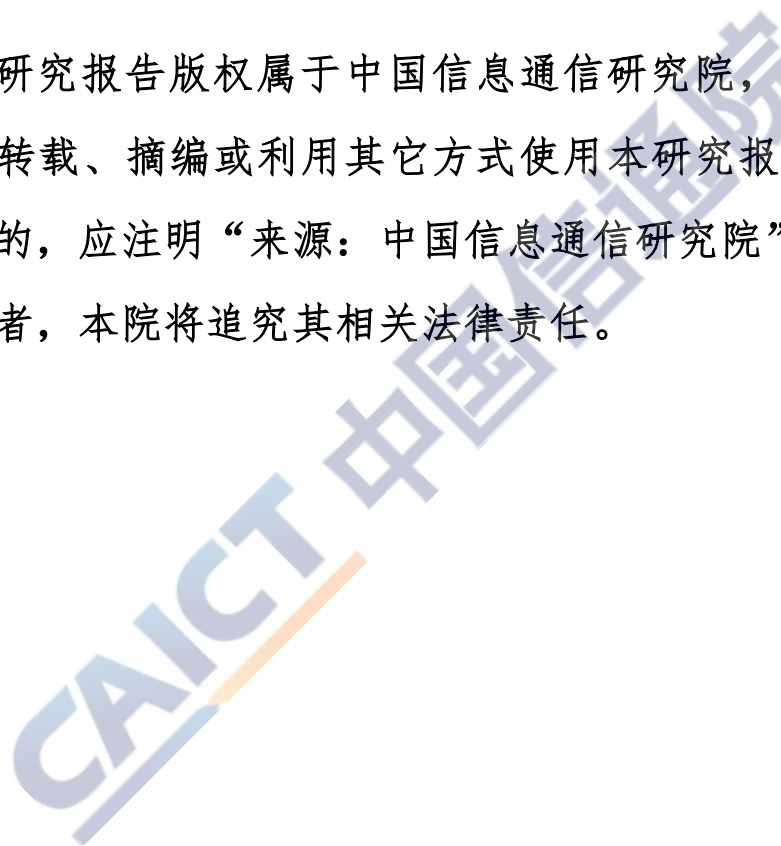


开源软件知识产权 风险防控研究报告 (2019 年)

中国信息通信研究院
2019年7月

版权声明

本研究报告版权属于中国信息通信研究院，并受法律保护。转载、摘编或利用其它方式使用本研究报告文字或者观点的，应注明“来源：中国信息通信研究院”。违反上述声明者，本院将追究其相关法律责任。



前 言

近几年，开源热度急剧攀升，已经成为信息通信技术创新的重要手段。开源软件不仅在云计算、大数据、人工智能、区块链等新兴技术发展中占据重要地位，在操作系统、数据库等基础软件的影响力也不断扩大。随着整个软件供应链开源化趋势越来越明显，开源软件的知识产权纠纷也会越来越多。开源软件有着独特的产权模式，其知识产权问题也较为复杂。企业使用开源软件面临知识产权风险有哪些、开源软件知识产权风险受哪些因素影响、如何规避知识产权风险等一系列问题亟待解决。

《开源软件知识产权风险防控研究报告》从开源软件生态及开源软件产权、开发、商业三种模式总结入手，阐述了企业使用开源软件面临的知识产权风险，并多角度分析开源软件知识产权风险影响因素，最后提出了企业开源知识产权风险防控建议。

CAICT 中国信通院

目 录

一、开源软件概述.....	1
(一) 开源软件定义及发展趋势.....	1
1. 开源软件定义.....	1
2. 开源软件产业发展趋势.....	4
(二) 开源软件生态.....	5
(三) 开源软件模式.....	8
1. 开源软件产权模式.....	8
2. 开源软件开发模式.....	9
3. 开源软件商业模式.....	10
二、开源许可协议知识产权规定分析.....	12
(一) 开源许可协议的本质及分类.....	12
(二) 常用开源许可协议知识产权规定介绍.....	13
1. GNU General Public License.....	14
2. Mozilla Public License.....	16
3. Apache License.....	18
4. BSD License.....	20
(三) 开源许可协议兼容性.....	21
1. 开源许可协议兼容的本质.....	21
2. 不同类型开源许可协议之间的兼容性.....	22
三、开源软件知识产权风险.....	24
(一) 开源软件知识产权风险种类.....	24
1. 版权侵权风险.....	24
2. 专利侵权风险.....	25
3. 商标侵权风险.....	26
(二) 开源软件知识产权风险影响因素.....	26
1. 开源许可协议.....	27
2. 开源社区知识产权管理.....	28

3. 开源软件的使用方式	28
四、企业开源知识产权风险防控建议.....	29
(一) 增强开源风险意识，建立风险防控体系	30
1. 正确认识企业开源知识产权风险，增强风险防范意识.....	30
2. 建立开源风险防控体系，加强风险防范工作协同.....	30
(二) 建立开源软件风险防控全流程管理	31
1. 开源软件引入时，综合考虑开源软件风险.....	31
2. 软件开发过程中，做好开源代码跟踪管理.....	31
3. 开源产品发布前，做好产品审核.....	32
(三) 法务部门需做好开源相关法律研究工作	32
1. 准确解读开源许可协议规则.....	32
2. 跟踪开源相关司法案例.....	32
附件：开源知识产权热点事件分析.....	33
(一) ARTIFEX 诉 HANCOM.....	33
1. 事件基本情况.....	33
2. 事件解读.....	33
(二) Facebook 修改开源许可协议.....	34
1. 事件基本情况.....	34
2. 事件解读.....	35
(三) MongoDB 变更开源许可协议.....	36
1. 事件基本情况	36
2. 事件解读	37

一、开源软件概述

近几年，开源软件发展呈现新趋势，开源软件生态不断完善，开源软件模式逐渐清晰。

（一）开源软件定义及发展趋势

1. 开源软件定义

开源软件有广义和狭义之分。广义的开源软件是指源代码公开，并且允许任何人学习、复制、修改、重新发布的计算机软件。狭义的开源软件特指符合开源软件促进会（OSI, Open Source Initiative）定义的开源软件。

OSI 对开源软件的定义包含 10 项标准¹：

——可以自由再发行 Free Redistribution：

开源软件的许可协议不应限制任何个人或团体将包含该开源软件的广义作品进行销售或者赠予。许可协议不能要求收取任何和这种销售相关的版权授权费或其他费用。

——直接访问源代码 Source Code：

开源软件的程序必须包含源代码，必须允许发布源代码及编译后的程序。如果产品中不包含源代码，那么必须提供一个公开的获取源代码的方式。这种方式可以收取的费用不能超过对源代码进行一次复制所需要的合理的成本，最好可以通过互联网提供免费下载。源代码的形式必须易于程序员修改，不能故意对源代码进行模糊化处理，

¹蔡俊杰等. 开源软件之道: ISBN 978-7-121-10483-1:[M]. 北京: 电子工业出版社, 2010.4.

也不得以预处理器或转译器输出的中间结果的形式提供源代码。

——衍生作品 Derived Works:

开源软件的许可协议必须允许修改和产生衍生作品，并且允许使用原有软件的许可协议发布它们。

——作者源代码的一致性 Integrity of the Author's Source Code:

只有在允许补丁文件和原有源代码一起发布的情况下，开源软件的许可协议才可以限制源代码以修改过的形式发布。许可协议必须明确地允许发布由修改后的源代码构建出的软件。许可协议要求衍生作品使用不同于原有软件的名称或版本号，以区别于原有软件。

——不歧视任何个人或团体 No Discrimination Against Persons or Groups:

开源软件的许可协议不能歧视任何个人或团体。

——不歧视任何特定用途 No Discrimination Against Fields of Endeavor:

开源软件的许可协议不能限制任何人把程序使用于某个领域，比如，不得规定软件不能用于商业目的或应用于遗传研究领域。

——许可协议的分发 Distribution of License:

程序所带的权利必须适用于所有接收方，而这些接收方无须执行附加的许可协议。

——许可协议不得限于某个特定产品 License Must Not Be Specific to a Product:

程序所带的权利与程序是否成为特定软件的一部分无关。如果某程序从特定软件中抽取而来并遵守程序本身的许可协议，那么该程序的所有接收方获得的权利与原有特定软件所赋予的对应部分的权利相同。

——许可协议不得限制其他软件 License Must Not Restrict Other Software:

开源软件的许可协议不能对同该许可协议下的软件一起发布的其他软件有任何约束。例如，开源软件的许可协议不能要求在同一媒介下发布的其他程序也必须是开源的。

——许可协议必须保持技术中立 License Must Be Technology-Neutral:

许可协议的条款不应指定任何的技术或接口。

OSI 定义的开源软件得到了业界的广泛认可，已成为开源运动的事实标准。OSI 可以对“许可协议符合开源定义”进行认证。只有通过认证的许可协议进行发布的软件才能使用“Open Source”的商标。经过 OSI 认证的开源许可协议包括：BSD-2-Clause、Apache v2、GPL v2、GPL v3 等 80 余种。除此之外，很多开源软件使用的开源许可协议未经过 OSI 认证。例如，BSD-4-Clause、SSPL、Apache2.0+Commons Clause 等。这些开源许可协议可能在常用许可协议中修改某条款或者增加某些特殊限制条款。由于未经 OSI 认证的开源许可协议数量众多、条款规定纷繁复杂，风险不确定性非常大，所以本研究报告主要聚焦于讨论狭义的开源软件。

业界很容易把“开源软件”和其他类型的软件进行混淆，以下对常见的误解进行澄清：**一是，源代码公开的软件不一定是“开源软件”**。微软的部分产品在全球推出了政府安全计划源代码协议，但用户只能看到源代码，不能修改、使用，因此不是开源软件。判断一个软件是否为开源软件，关键是看其软件许可协议如何规定软件的使用、发布、复制和演绎等整个过程。**二是，开源软件不是公有软件**。公有软件明确声明放弃版权或者版权中的经济权利有效期届满。但是，开源软件受版权保护，不能任意使用。**三是，开源软件不等同于免费软件**。开源软件通常是免费的，但不是所有的免费软件都是开源软件。**四是，自由软件是开源软件的子集**。自由软件特指以 GPL 类许可协议发布的软件，定义比开源软件更为严格。自由软件要求确保使用者能获得使用软件的真正自由，不允许修改源代码后闭源出售软件。很多 OSI 认可的开源软件并不符合自由软件的定义。

2. 开源软件产业发展趋势

近几年，“软件即服务”的概念深入人心，软件脱离了传统意义上出售许可的模式，IT 产业逐渐向服务化转型。开源软件的盈利模式与此趋势相吻合。此外，开源秉承开放的理念，能广泛汇集产业链上下游的力量。因此，开源对 ICT 产业的影响力不断扩大，其发展呈现以下新趋势：

一是，开源软件在软件产品体系、供应体系渗透率不断提高，已经成为软件生产链条中必不可少的要素。在操作系统、数据库等基础软件领域，开源影响力逐年提升。在云计算、大数据、人工智能、工

业互联网等新兴领域，开源已经成为主要开发模式，例如，超过 80% 的企业会选择开源软件部署私有云。系统软件、应用软件、设备固件等越来越多采用开源代码，这使得开发、编译、测试的整个软件供应链呈现开源化趋势。开源软件对软件产业的影响由量变到质变，成为软件企业难以绕过的“生产要素”。

二是，科技巨头纷纷拥抱开源，加大对开源的投入力度。一方面，科技巨头加大对开源代码的贡献力度。2018 年，在全球最大的托管网站 GitHub 上拥有开源贡献者数量最多的前 10 家机构中，微软、谷歌、英特尔、Facebook 排名分别为第一、第二、第五、第七。另一方面，科技巨头对开源生态的重视程度日益提高，控制手段不断加强，逐步通过高额收购开源资产等方式加大对开源社区的影响。如 2017 年，谷歌收购机器学习开源社区 Kaggle；2018 年，微软收购全球最大的代码托管平台 GitHub。

三是，开源社区与标准组织合作越来越频繁，开源对标准制定的影响越来越大。技术创新在开源社区与标准组织中同步进行，开源项目开发与标准制定相互促进。例如，ONAP 开源社区设立了标准协调员，负责定期组织 ONAP 与行业标准组织的联合研讨会议，以收集当前社区开发工作中遇到的标准协作意向与问题，并同时征求来自标准组织的反馈意见。

（二）开源软件生态

开源软件生态包括外部环境、开源基金会、开源社区、开发者、开源贡献方、开源参与方、开源项目托管网站等组成部分，如图 1 所

示。

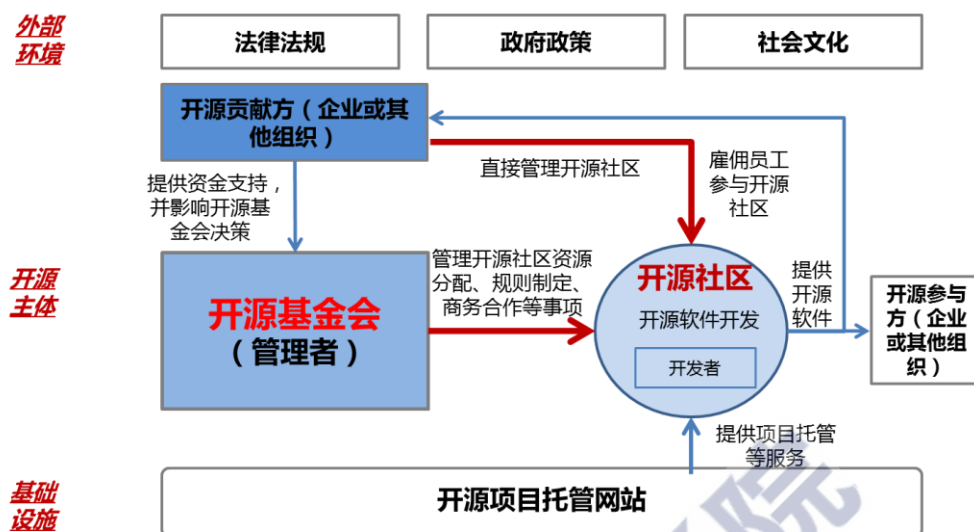


图1 开源社区运作示意图

● 外部环境

开源生态的发展会受所在国法律、政策、文化环境的影响。国家进出口管制、知识产权等相关法律法规会约束各国开源主体的活动。政府支持开源软件发展的政策、社会开放创新的文化氛围等，也能有利于开源社区的发展。

● 开源社区

开源社区是依托网络平台建立，以共同开发开源软件为目的，由诸多开发者形成的虚拟组织。开源社区向开源贡献方、开源参与方提供开源软件。开源社区的管理者可以是开源基金会、企业或高校等其他组织。

● 开发者

开发者是指参与开源软件开发的程序员。

● 开源基金会

很多大型开源社区是由开源基金会（如 Open Stack、Apache、Linux 等）管理。开源基金会作为开源社区的管理者，把控开源社区资源分配、组织方式、规则制定、知识产权等的决策权。开源基金会资金主要来源于企业捐赠。企业通过进入开源基金会的董事会的方式来影响开源社区的发展。

● 开源贡献方

开源贡献方是指向开源社区做出贡献的企业或高校、科研院所等研究机构。有些开源贡献方直接管理开源社区而不是交由开源基金会管理。例如，Google 直接管理 Android、TensorFlow 等开源社区，Facebook 直接管理 Buck、React Js、Presto 等开源社区。除进行社区管理之外，开源贡献方对社区的贡献方式主要有两种：一是，直接雇佣员工参与开源社区，贡献代码；二是，通过向开源基金会提供资金间接影响开源社区的发展。

● 开源参与方

开源参与方是指未对开源作出实质贡献，仅仅使用开源软件的企业或其他组织。开源参与方只能是开源社区的跟随者，对开源技术发展没有话语权。

● 开源项目托管网站

开源项目托管网站是开源软件开发者进行开发管理的集中场所，为开源项目提供存储、协作和发布的平台，提供代码存储、文档管理、版本控制、问题跟踪、信息交流等服务。开源项目可以根据需要选择

或更换项目托管网站。

（三）开源软件模式

开源软件的产权模式、开发模式、商业模式密切相关、相互影响。开源软件的产权模式是保证开源社区进行软件开发的基础，开发模式保障产品研发流程及质量，商业模式是企业走开源之路的持续源动力。

1. 开源软件产权模式

无论是闭源软件²还是开源软件，都受版权法保护。版权保护不需要履行注册手续，自产生之时就自动获得版权。软件的程序及相应的文档，都属于版权保护的客体。版权法授予软件版权所有者人身权和财产权，人身权包括署名权、发表权、保护作品完整权和修改权，财产权包括复制权、发行权、出租权、汇编权、翻译权等。

开源软件、闭源软件都通过软件许可的方式将版权法授予的权利许可给用户，二者版权授权相同点是：通常都不包括人身权。大部分开源许可协议都规定，用户在分发软件时应当尊重原作者的版权人身权，例如保留版权声明、不得擅自使用贡献者的姓名和商标宣传推广等。

开源软件和闭源软件版权授权的差异主要在财产权授权方面。闭源软件对财产权的授权多通过有偿方式实施，授权内容也会有所限制。开源软件许可协议对财产权部分一般持免费和充分授权的态度。开源软件版权所有者公开程序的源代码，并允许被许可人在遵守开源许可

² 闭源软件（Closed source software）：指源代码的获取收到特定限制的软件，是与开源软件相对立的一个概念。

协议的前提下，阅读、修改、复制、再发布程序，实际上向被许可人让渡了软件作品的复制权、修改权等财产性权利³。

2. 开源软件开发模式

开源软件不仅仅在版权授权方式上有很大创新（如上节所述），在开发模式上也有很大突破。传统闭源软件采用“大教堂”模式，有着严密的管理和封闭的集中式结构。开源社区采用“集市”模式，是一种过程透明的开放式、分布式的软件开发模式。

开源软件开发过程如图 2 所示。首先，项目发起方发起一个新的**开源项目**。项目发起方可以是开源基金会、企业、高校、研究机构或者个人。项目发起方主要做好两方面准备工作：**一是，贡献初始代码，对项目有初步设计**，并在后续项目中起到关键作用，通常是项目的领导者；**二是，确定开源社区的相关规则**。开源社区活动遵循规则包括开源社区运作规则和开源许可协议。开源社区运作规则是开源社区的“规章制度”，对开源社区管理作出相关规定，主要包括：贡献者进（出）规则、领导角色确定、决策机制、知识产权保护等规则。开源许可协议是开源软件的“使用规则”，规定了用户使用开源软件的权利和义务。

项目发起之后，世界各地、各个企业不同的开发者在遵守社区运行规则、开源许可协议的前提下，参与开源软件开发，并通过网络、会议等方式进行交流。开源社区中，开发者有不同的角色，拥有的权利和承担的义务不相同。例如，在 Apache 社区中，项目开发人员分

³张平，马骁. 共享智慧——开源软件知识产权问题解析: ISBN 7-301-09798-0:[M]. 北京: 北京大学出版社, 2005,12.

为用户、开发者、提交者、PMC 成员等角色，提交者、PMC 成员都是从开发者中一层层通过选举模式选拔上来的。用户以补丁报告和建议的方式提供反馈，项目开发者可以提交代码或者文档，项目提交者可以将代码写入程序，项目管理委员会成员可以直接提交代码到代码仓库。

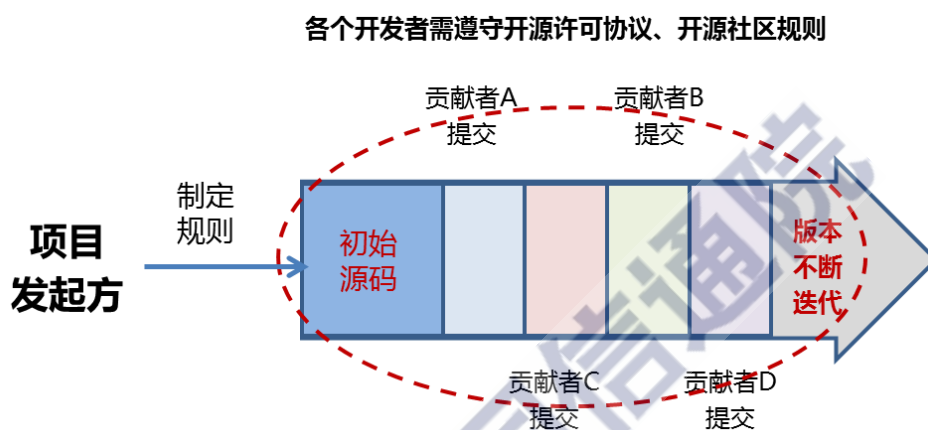


图 2 开源软件开发过程

3. 开源软件商业模式

开源软件可以用于商业用途，其商业模式与闭源软件有所不同。闭源软件一般通过控制软件源代码、出售软件许可来获益。开源软件商业模式多样，但需注意与开源许可协议的协同。目前盈利模式主要有以下五种：

- 双重授权

双重授权是指针对个人/商用进行不同授权或不同版本（开源社区版本、企业版本）进行不同授权。开源社区版本以开源许可协议免费许可给用户，便于测试软件、获得改进信息、赢得口碑。企业版本采用商业许可，通过为企业使用者提供更丰富的功能以及提供技术支

持、担保服务等方式来盈利。该模式主要适用于软件服务商，前提条件是公司拥有开源软件所有代码的版权。采用这种模式通常使用 GPL、AGPL 等强著佐权 (Copyleft) 许可协议，如 MySQL、MongoDB 等软件。

- 通过硬件捆绑盈利

硬件捆绑是指利用免费而优质的开源软件带动它所属的硬件的销售。该模式主要适用于硬件设备商，任何许可协议的软件都可以采用这种方式，例如 IBM 等服务器供应商通过捆绑免费的 Linux 操作系统销售硬件服务器。

- 通过出售增值产品盈利

出售增值产品盈利是指向开源软件的商业用户贩卖增值服务或者增强组件、开发工具等许可，通过附加更多的闭源软件来增加收入。该模式适用于提供基础软件的服务商，通常会选择较为宽松的许可协议，如 Android 系统。谷歌通过 AOSP（安卓开源项目）汇聚人气，以开源方式为谷歌开发者联盟成员提供基础服务，包括基础 Linux 内核、Dalvik 虚拟机的 Android 基础框架代码，以及部分应用和 API。在 Android 系统之上的 GMS（谷歌移动服务）则采用闭源方式来获得盈利，包括谷歌移动互联网的核心应用以及关键 API，如搜索、邮件、日程、地图、街景等服务。

- 通过技术支持盈利

技术支持来盈利是指产品免费，但是通过产品的技术咨询、维护修理、技术文档、人员培训等服务额外收费。该模式适用于软件服务商，对许可协议没有要求，但 GPL 许可协议的软件通常采用这种方式，

如红帽公司的开源产品大都采用这种盈利模式。

- 通过广告业务盈利

通过广告业务盈利是通过内嵌广告而达到盈利的目的，而产品本身需要做的只是如何吸引更多用户。该模式适用于互联网厂商，可采用任何许可协议，如火狐公司推出的开源浏览器。

二、开源许可协议知识产权规定分析

理解开源许可协议是认识开源软件知识产权问题的关键之所在，因此，接下来将重点介绍开源许可协议，包括许可协议本质及分类、知识产权规定以及开源许可协议兼容等问题。

（一）开源许可协议的本质及分类

每个开源软件发布时，都会附一个开源许可协议。开源许可协议将特定的权利赋予用户，同时也会规定用户使用开源软件时必须遵守的约束。常用的开源许可协议包括：Apache-2.0、BSD、GNU GPL、GNU LGPL、MIT、MPL、CDDL、EPL 等。

从已有的司法判决来看，无论是 2006 年发生在德国的 Welte 诉 D-link 案，还是 2017 年 ARTIFEX 诉 HANCOM 案件中，法院都认可开源许可协议为许可合同。具体来讲，**开源许可协议是涉及版权、专利、商标等一系列权利义务的格式合同，且自动生效。**

开源许可协议赋予用户的权利和义务不尽相同，按照源代码再分发的规定可以将开源许可协议分为三大类：

- 强著佐权型许可协议

强著佐权型许可协议明确修改版本须以同一许可协议发布。如果许可协议要求一个软件包含该协议下部分代码，完全发布时必须作为整体适用该协议，则该协议为强著佐权型许可协议，如：AGPL、GPL许可协议。对于遵循强著佐权型许可协议的开源软件，企业不能进行二次开发并闭源出售软件许可。

- 弱著佐权型许可协议

如果许可协议规定，一个软件包含该协议下部分代码，完全发布时某些部分必须适用该协议，其他部分可在其他协议下发布，则该协议为弱著佐权型许可协议，如：LGPL、MPL、CPL、EPL等。对于遵循弱著佐权型许可协议的开源软件，企业可以向开源代码添加闭源组件，使之商品化，并对外销售。

- 宽容型许可协议

宽容型许可协议对已修改代码的许可方式没有任何要求，如：BSD、Apache、MIT许可协议等。对于遵循宽容型许可协议的开源软件，企业可按照自己的需求，把自己开发的软件产品重新以商业许可协议或其他兼容的许可协议发布。

从强著佐权型许可协议到弱著佐权型许可协议，再到宽容型许可协议，商业友好度是递增的。但是，企业不能仅仅以商业友好度来判断开源许可协议知识产权风险的大小。有些宽容型许可协议，比如BSD、MIT，没有明示专利许可，专利侵权风险反而很大。不同的开源许可协议规定不同，合规的难易程度不同，知识产权风险点也不同。

（二）常用开源许可协议知识产权规定介绍

研究报告根据开源许可协议的分类并结合其使用频率,选取了 GNU GPL、MPL、Apache、BSD 四个开源许可协议进行详细分析。

1. GNU General Public License

(1) 许可协议基本情况

GPL 许可协议是自由软件创始人 Richard Stallman 为其 GNU 项目撰写,目前由自由软件基金会管理。开源许可协议的管理者有修改许可协议、发布许可协议新版本的权利。1989 年 1 月,GPL v1 发布;1991 年 6 月,GPL v2 发布;2007 年 6 月,GPL v3 发布。目前,比较常用的版本是 GPL v2、GPL v3。

GPL 是强著佐权型许可协议,具有传染性。GPL 许可协议的程序的演绎作品也需要以 GPL 许可协议发布,不允许修改后和衍生的代码作为闭源商业软件发布和销售。使用 GPL 许可协议的著名开源软件包括:Linux 内核、MySQL 数据库、OpenJDK 平台等。

下文主要详细解读 GPL v3。

(2) 版权规定

GPL 许可协议中,许可人授予被许可人享有的最主要的权利包括:一是,自由地运行 GPL 许可下的程序;二是,自由地获得源代码;三是,自由地复制、发布程序的源代码;四是,自由地修改程序,并将修改过的程序或基于程序的作品发布。

被许可人在获得相关权利的同时,也需要履行相应的义务。GPL 许可协议对未修改源代码的程序复制发布、修改源代码形成的程序复制发布、以目标码或可执行程序形式复制或分发程序这三种情况做了

明确规定。

复制、发布未修改的程序源代码完整副本时，需要提供许可协议全文、版权声明等相关声明。许可协议明确说明，复制发布者可以免费或付费发布副本，也可以就提供支持或担保而收费。

发布修改过的源代码时，必须将整个软件作为一个整体以 GPL v3 许可协议向获取副本的人发布。不管软件中的各个部分如何打包在一起，GPL v3 对软件的所有部分适用。但是，如果覆盖程序和其他可区分且独立的程序组合成了“聚合体”，则不会导致对聚合体的其他部分适用 GPL v3。该条款是 GPL 许可协议“传染性”的核心。

以目标码或可执行程序形式复制或分发程序时，除了遵循相关的声明义务，还需要以许可协议规定的方式发布对应的源代码。目标代码中可分离的部分，只要其源代码属于系统库，不包含在对应源代码中，就不需要随目标代码一同发布。

（3）专利规定

GPL v3 明示了专利许可：任何贡献者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。

（4）商标规定

GPL v3 没有明确的商标规定，无商标使用授权。

（5）其他重要规定

GPL v3 有不担保规定：版权所有者和/或其他第三方“按原样”提供程序，没有任何形式的担保，无论是明示的还是默示的。任何版

权所有者，修改和/或发布程序的第三方对程序使用者造成的损失不负任何责任。

GPL v3 明确规定了违约责任。如果没有按照许可协议传播或修改覆盖程序，权利将自动终止。但是，GPL v3 对违约者的态度更为温和，如果在相关期限内修正违反行为，则可以重新获得许可。

2. Mozilla Public License

（1）许可协议基本情况

MPL 许可协议是 Mozilla 基金会推出并广泛使用于其旗下软件的开源许可协议。MPL v1 于 1998 年初发表，是 Netscape 的 Mozilla 小组为其开源软件项目设计的许可协议。MPL 许可协议出现的最重要原因是，Netscape 公司认为 GPL 许可协议没有很好地平衡开发者对源代码的需求和他们利用源代码获得的利益。MPL v1 之后又发布了 MPL v1.1，目前最新版本是 MPL v2。

MPL 是弱著佐权型许可协议，比 GPL 类许可协议宽松。MPL 仅要求 MPL 许可的那部分代码及修改遵循 Copyleft 机制，增添的新文件可以闭源。使用 MPL 开源许可协议的著名软件包括：Mozilla Firefox 浏览器、Mozilla Thunderbird 邮件客户端、Adobe Flex 平台等。

以下主要介绍 MPL v2.0 的相关规定。

（2）版权规定

贡献者授予使用者全球性的、免费的、非排他性的许可：在贡献者许可的版权范围内，使用者可以使用、复制、提供、修改、展示、运行、发布或以其他方式使用其贡献。

以源代码形式发布覆盖软件时，必须告知接受者该覆盖软件的源代码形式是由 MPL v2 许可协议规定管理的，并告知其如何获得此许可协议的副本。接受者不得修改或限制接受者对源代码形式的权利。

以可执行形式发布覆盖软件时，必须以合理的方式告知该覆盖软件可执行形式的接受者如何获得该覆盖软件的源代码，并按照要求提供源代码。程序接受者可以依据 MPL v2 发布可执行形式，也可以在不同的协议条款下再许可该可执行形式，只要可执行形式的许可协议不限制或改变接受者依据本许可协议享有的对源代码形式的权利。

程序接受者可以根据您选择的条款创建和发布一个“更大的作品”，只要您也同时遵守本许可协议对覆盖软件的要求。

（3）专利规定

MPL 明示专利许可：贡献者授予使用者全球性的、免费的、非排他性的许可，允许使用者根据贡献者的权利要求，制造、使用、销售、许诺销售、使他人制造、进口及转让其贡献或其贡献者版本。

MPL 存在“专利报复”条款：如果接受者对任一主体发起专利侵权诉讼（不包括宣告式判决，反诉和交叉诉讼），声称一个贡献者版本直接或间接侵犯专利权，那么任一和所有覆盖软件的贡献者根据本许可协议授予其的权利都将被终止。

（4）商标规定

MPL 不授予任何贡献者商标、服务标记或徽标（除可能需要符合有关声明要求的情况外）许可。

（5）其他重要规定

MPL 存在不担保条款：在本许可协议下提供的“覆盖软件”是基于如下一如既往的基础，不承担任何形式的担保责任，无论是明示、暗含或法定的担保。在任何情况或任何法律理论下，不论是侵权（包括过失侵权）、合同或其他，任何贡献者或任何“覆盖软件”的发布者都不会为了任何直接的、间接的、特殊的、偶然的或任何性质的结果性损害承担责任。

MPL 规定了“司法管辖”：任何与 MPL 许可协议有关的诉讼只能在被告主营业务所在地的司法管辖区提出，而且这些诉讼应当由该司法辖区的法律管辖，而不必考虑法律冲突条款。

3. Apache License

（1）许可协议基本情况

Apache 许可协议是由 Apache 基金会管理。它基于 BSD 和 MIT 许可协议，增加了一些重要条款。Apache 许可协议共有三个版本：Apache v1、Apache v1.1 和 Apache v2。目前使用最多的是 Apache v2，该版本是 2004 年发布的。

Apache 许可协议是宽容型许可协议，允许后续使用者在 Apache 代码基础上进行二次开发并闭源出售。Apache 许可协议商业非常友好，因此被广泛使用，Apache 基金会下所有项目都使用 Apache 许可协议，如 Apache HTTP、Tomcat Web、Struts 框架等。

（2）版权规定

每个贡献者授予用户永久性的、全球性的、非排他性的、免费的、不可撤销的版权许可协议，以源程序形式或目标形式复制、开发演绎

作品、公开展示、公开运行、进行分许可、以及发布作品和上述演绎作品。

用户复制和发布作品或演绎作品时，无论是否修订，无论是以目标程序还是源程序形式发布，都需要提供许可协议副本，修改声明，先前作品所有的版权、专利、商标的权属声明等文件。

（3）专利规定

Apache v2 明示了专利许可：每个贡献者授予用户永久性的、全球性的、非排他的、免费的、无许可费的、不可撤销的（除非在本部分另行声明）专利许可证，以允许使用者制造、让人制造、使用、许诺销售、销售、进口和以其它方式转让相关软件。

Apache v2 存在“专利报复”条款：如果用户就作品或作品中所涉及的贡献起诉，指控其构成直接或间接侵权（包括交诉或反诉），那么根据本许可协议，授予用户针对作品的任何专利许可证将在提起上述诉讼之日起终止。

（4）商标规定

Apache v2 限制商标使用：许可协议并未授予用户使用许可协议颁发者的商号、商标、服务标记或产品名称，除非是为以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。

（5）其他重要规定

Apache v2 有不担保规定：许可协议颁发者遵循“按原样”的原则提供作品，而不附加任何明示或暗示的保证或条件。贡献者不对用户因使用本许可协议而受到的损失负责。

4. BSD License

（1）许可协议基本情况

BSD 许可协议由加利福尼亚大学伯克利分校管理。BSD 许可协议有多个变种，统称为 BSD 风格的许可协议。最初的 BSD 许可协议在 20 世纪 80 年代末随操作系统 BSD 的 Net/1 版推出，现在被称为原版 BSD 许可协议、旧版 BSD 许可协议或 4 条 BSD 许可协议。1999 年，加利福尼亚大学伯克利分校将原版 BSD 许可协议中的广告条款删除，从而产生了新版 BSD 许可协议，又被称为修改版 BSD 许可协议、新的简化的 BSD 许可协议、3 条款 BSD 许可协议等。3 条款 BSD 许可协议、2 条款 BSD 许可协议是经过 OSI 认证的。

BSD 许可协议是宽容性许可协议，对已修改代码的许可方式没有任何要求，便于企业在此基础上修改或二次开发。使用 BSD 许可协议的著名开源软件包括 FreeBSD 操作系统、PostgreSQL、JForum 等。

以下主要介绍三条款 BSD 许可协议。

（2）版权规定

无论是否修改，再发布源代码时，必须保留先前版权声明、许可协议原文和免责声明。

无论是否修改，再发布二进制形式的代码时，必须附加版权声明许可协议原文以及免责声明和/或发布时所提供的其他材料的副本。

（3）专利规定

BSD 许可协议没有专利相关条款，没有明示专利授权。一方面，版权所有者、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，程序接收者没有获得版权所有者、贡献者关于程序包含的专利的授权，既不向后续使用者提供知识产权担保，也没有“退出机制”要求其停止发布“不清洁”软件，因此一旦原程序存在侵权，后续构成演绎作品的程序专利侵权风险也会较大。

（4）商标规定

在发布软件时，不能将版权所有人或者贡献者的姓名用于宣传、推广软件。如果需要，必须先获得版权所有人或者贡献者的明确的书面许可；否则，版权所有人或者贡献者有权提起商标侵权诉讼。

（5）其他重要规定

BSD 许可协议存在不担保规定：版权所有人和贡献者“按原样”提供本软件，不提供任何明示或者暗示的担保。在任何情况下，版权所有者或者贡献者均不对本软件任何方式产生的任何损失承担任何法律责任。

（三）开源许可协议兼容性

1. 开源许可协议兼容的本质

开源许可协议众多，仅 OSI 认证的开源许可协议就达 80 余种，此外还存在大量没有经过 OSI 认证的所谓的“开源许可协议”。不同开源许可协议在概念界定、版权许可、专利许可、免责等方面的规定有很大不同，难免出现不同许可协议的部分条款存在冲突的情况。当程序开发人员把不同开源许可协议许可的代码组合成一个大程序发布时会产生很多问题：例如，形成的大程序最终以哪个许可协议发布，

许可协议条款之间的冲突造成被许可人利益受损该如何解决等；这就是许可协议兼容性问题。从法律层面来讲，不兼容的许可协议许可的开源代码不能合并使用到一个程序中。

目前，业界对兼容性的理解并不统一。广义的兼容性是从“组合作品”角度来考虑。如果两个（或多个）开源许可协议的许可条款规定不冲突，可以将这些不同开源许可协议下的代码组合在一起形成一个可以发布的软件，那么就可以说这两个（或多个）开源许可协议是互相兼容的。狭义的兼容性是从“演绎作品”角度来考虑，不仅仅考虑不同许可协议下的代码是否可以组合在一起，还需要考虑组合之后形成的组合作品以哪个许可协议发布。

许可协议兼容性问题本质上是不同许可协议许可要求的冲突，两个许可协议中的强著佐权型一方的许可规定并不能完全包含另一方的规定或与另一方出现明显条款上的冲突。但是，两个协议之间轻微的区别不影响兼容性问题的判断，例如 MPL v2 和 EPL v1 之间关于专利报复条款的范围有细微不同，但不影响两个协议之间的兼容。

在处理许可协议兼容性问题时，需要明确：只有将不同许可协议下的开源组件组合成一个软件并且需要发布时才需要考虑许可协议兼容性的问题。如果不涉及软件发布，只是公司内部使用不同开源组件的话，不需要考虑许可协议兼容性。因为许可协议中大部分限制条款都是针对软件传播行为的。

2. 不同类型开源许可协议之间的兼容性

(1) 强著佐权型许可协议之间的兼容性

目前常见强著佐权型协议主要有 GPL v2、GPL v3 和 AGPL v3。由于强传染性许可协议都要求再发布时需遵循各自的许可协议，所以一般情况下强著佐权型协议之间很难兼容，例如 GPL v2、GPL v3 是不兼容的。但是，GPL v3、AGPL v3 的第 13 条中都规定了这两个许可协议一起使用的情况，所以 GPL v3、AGPL v3 是兼容的。

（2）强著佐权型和弱著佐权型许可协议之间的兼容性

强著佐权型与弱著佐权型许可协议之间的兼容性非常棘手。开源软件选择强著佐权型许可协议的主要原因是：在原始软件基础上修改得到的新软件必须是在同样的强著佐权型许可协议下可获得。此举主要是为了防止将开源软件变为闭源软件，但同时也阻碍了开源软件以弱著佐权型许可协议发布。GPL 许可协议要求“用户必须以该许可协议的许可条款发布整个程序的源代码”，MPL 1.1 要求修改作品“受本许可协议的条款管辖”。这两个规定分别是强著佐权型和弱著佐权型许可协议的核心特征，但这些规定是相互矛盾的。任何两个强著佐权型和弱著佐权型的许可协议都会面临相似的问题。

（3）强著佐权型和宽容型许可协议之间的兼容性

大部分强著佐权型许可协议与宽容型许可协议之间是兼容的；但是，当宽容型许可协议的特殊条款和强著佐权型许可协议有明显冲突时，也会存在不兼容的情况。例如，GPL v2 许可协议第 6 条中规定“您不可以就本许可协议赋予接收人的权利附加任何进一步的限制”，然而 Apache v2 许可协议中有专利报复条款，两个条款的明显冲突造成两个许可协议不能兼容。

（4）弱著佐权型许可协议之间的兼容性

弱著佐权型与弱著佐权型协议之间的兼容性问题较为模糊，版权人并未给出明确的答复，但是按照业内共识一般情况下是兼容的。

（5）弱著佐权型和宽容型许可协议之间的兼容性

按照业内共识，弱著佐权型和宽容型许可协议之间一般都是兼容的。

三、开源软件知识产权风险

开源软件已成为 ICT 巨头竞争的重要战场，未来商业纠纷可能会愈加频繁。由于开源软件模式的特殊性，其知识产权问题较为复杂，需要理清开源软件知识产权风险的种类及风险来源。

（一）开源软件知识产权风险种类

1. 版权侵权风险

使用开源软件的版权侵权风险主要来源于两个方面：

一是，开源软件使用者没有按照开源许可协议的规定使用开源软件，从而导致版权侵权。每一个开源许可协议都详细规定了开源软件使用者的权利和义务，开源软件使用者需完全按照许可协议要求进行复制、分发，如提供版权声明和源代码、标注修改时间等。如果使用者没有按照规定使用开源软件，例如，使用 GPL 许可协议的代码再发布软件时没有提供源代码，就属于违约行为，会导致开源软件版权人通过许可协议赋予被许可人的权利被收回，那么，该使用者对开源软件的复制、修改、再发布就会造成版权侵权。

二是，贡献者将自己不具有版权的代码贡献到开源社区，使得开源软件本身存在版权瑕疵。开源软件的开发模式决定了会有来自世界各地、不同职业的开发人员向开源软件贡献代码。如果开源社区没有经过严格的代码审核，贡献者可能会将其他软件的代码复制或修改后贡献到开源社区，导致开源软件“不清洁”。后续使用者即使完全按照许可协议规定使用该开源软件，也会侵犯他人版权。因为绝大多数开源许可协议都有“不担保”条款，开源软件的用户使用了“不清洁”的开源软件所造成的侵权责任，需要自己承担。

2. 专利侵权风险

使用开源软件的专利侵权风险来源于内部、外部两个方面：

内部专利侵权风险，是指开源软件的贡献者以个人名义对其中某项技术申请专利并向开源使用者发起专利诉讼。有些开源许可协议（如 Apache v2）明示了专利许可要求且存在专利报复条款，内部专利风险相对较小。然而，部分开源许可协议（如 BSD、MIT 等）没有明示专利许可规定，开源代码贡献者可以向开源代码使用者提起专利诉讼并收取专利许可费，因此内部专利风险较大。

外部专利侵权风险，是指不受开源许可协议约束的第三方向开源软件使用者发起专利诉讼，声称在开源程序中使用到其专利。例如，微软就曾表示过包括 Linux 在内的开源软件使用了微软的专利。不同于版权仅仅保护计算机软件创作的表达，专利能够保护具体的方法和功能，且享有绝对的排他权，在专利有效期内，权利人能够禁止实施专利技术。不仅是开源软件，闭源软件也存在专利侵权风险。此类风

险与开源许可协议无关，且难以规避。

3. 商标侵权风险

使用开源软件的商标侵权风险来源于两个方面：

一是，开源软件使用的开源许可协议未经 OSI 认证，但使用了 open source 商标。只有通过开源促进会认证的开源许可协议，才可以使用 OSI 关于 open source 的商标，未授权使用，就会带来商标侵权风险。

二是，开源软件使用者未经授权擅自使用贡献者的商标、商号、服务标记等进行软件宣传，导致商标侵权。绝大部分开源许可协议都不包括商标授权，有些许可协议明确规定不得使用商标。例如，Apache v2 规定：“本许可协议并未授予用户使用许可协议颁发者的商号、商标、服务标记或产品名称，除非是为以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。”因此，未经额外许可使用商标，会有很大隐患。

企业在使用开源软件时，还会面临由软件质量、信息安全、商业纠纷、技术垄断等引起的知识产权方面的风险。由于研究报告篇幅有限，在此不做详细论述。

（二）开源软件知识产权风险影响因素

从开源软件产权、开发、商业角度来看，开源许可协议、开源社区知识产权管理方式、企业开源软件的使用方式都会影响知识产权风险大小。

1. 开源许可协议

从产权角度来看，不同的开源许可协议赋予用户不同的权利和义务，版权、专利、商标等方面的规定不同，因此带来的风险也不同。

在版权方面，不同许可协议对于软件再发布时的许可方式、是否提供源代码、是否需要对修改做出标示规定不同。例如，GPL 许可协议要求：GPL 许可的程序的演绎作品整体发布时必须在 GPL 许可下进行发布，这就是所谓的 GPL 许可协议的“传染性”。Apache、MIT、BSD 则对演绎作品的发布方式没有要求。因此，使用 GPL 许可协议下的源代码在软件再发布时风险较大。

在专利方面，GPL v3、Apache v2 明示了专利授权并有专利报复条款，MIT、BSD 则无此类规定。GPL v3 在第 11 条中明确规定贡献者将程序相关的专利许可给接受者，第 10 条规定软件发布者不可以发起诉讼（包括互诉或者反诉）主张由于制造、使用、销售、批发或者引进程序或其任何一部分而侵犯了任何专利权。使用 GPL v3 许可下的代码，来自贡献者内部的专利诉讼风险就会比较小。但是，不管哪个开源许可协议，都无权约束第三方，因此即使是 GPL 许可下的软件，也会受到第三方软件专利的威胁。

在商标方面，开源许可协议通常只是源代码的版权授权，一般不包括商标授权，因此不能随意使用未经授权的商标。有些开源许可协议，如 Apache v2 对商标使用进行了明确规定，限制用户使用许可协议颁发者的商号、商标、服务标记或产品名称，那么在使用 Apache v2 许可的开源软件时，就需要非常注意商标使用规范。

在使用开源软件时，需注意以下两点：**一是，找出开源软件使用的许可协议及其版本号。**不同版本的许可协议可能存在差异，仔细阅读该版本的开源许可协议条款，严格按照条款规定使用开源软件。**二是，注意跟踪开源软件是否更改其许可协议。**开源社区管理者可能会由于各种原因随时更改开源软件使用的许可协议，甚至由开源变为闭源。如遇此情形，企业需重新评估开源许可协议风险之后再使用该开源软件。

2. 开源社区知识产权管理

从开发角度来看，开源社区的知识产权规定不同，管理严格程度不同，开源软件的知识产权风险大小也不同。

所有开源软件均面临版权瑕疵风险，风险大小与开源社区的管理和代码审核的规定相关。例如在 Apache 社区中，在一个开发人员成为提交者（committer）之前，他必须首先签署一份个人贡献者许可协议（ICLA）并备案，声明将所有贡献授权于 Apache 软件基金会，并保证这些贡献都是他们自己原创的工作，没有侵犯第三方的知识产权，有权作出如上授权。如果一个开发者自己并不拥有对自己工作内容的这种权利（例如，一个商业公司对雇员工作拥有权利），那么还需要商业公司的代表署一个企业贡献许可协议（CCLA）。因此，Apache 基金会所属的项目对用户来说由于版权瑕疵带来的侵权风险较小。

3. 开源软件的使用方式

从商业使用来看，企业对开源软件的利用方式不同，开源软件的

知识产权风险不同。

如果企业仅仅作为最终用户单纯地运行开源软件的可执行形式，那么使用开源软件的风险很小。如果企业修改了开源软件但是不发布，仅供企业内部使用，如搭建一个网站提供网络服务，这种情况开源软件的风险也很小（但是 AGPL 除外）。如果将开源软件原封不动或部分修改之后包含在自己的产品中进行再发布，这样的使用风险最大，企业在使用时必须遵守开源许可协议规定，对产品进行再发布。

四、企业开源知识产权风险防控建议

开源风险防控是个系统工程。开源知识产权风险防控是一件长期而持续的工作。企业要从增强风险意识、构建风险防控体系、加强流程管理、做好法律研究等方面防控风险，如图 3 所示。



标记 ★ 处表明：中国信息通信研究院知识产权中心可向企业提供相关内容的咨询或培训服务

图 3 开源知识产权风险防控工作

（一）增强开源风险意识，建立风险防控体系

1. 正确认识企业开源知识产权风险，增强风险防范意识

第一，企业要正确认识开源软件使用风险。一方面，企业使用开源软件必然会面临知识产权、开源软件质量、信息安全、开源技术垄断等一系列风险，知识产权风险是其中之一；另一方面，开源软件知识产权风险并非无限大，企业不应因为开源软件存在知识产权风险就放弃使用。**第二，企业需增强开源软件知识产权风险防范意识。**开源软件知识产权风险防控关系到企业业务发展，不仅仅是法务人员的职责，需要引起企业高层、全体技术开发人员的重视。企业可通过开展开源法律风险培训等方式，加深企业员工对开源软件的理解，增强企业开源风险防控意识。

2. 建立开源风险防控体系，加强风险防范工作协同

开源法律风险防控是一个系统工程，企业需协同各个部门建立开源法律风险防控体系。**一是，开源法律风险防控需要与公司发展战略、产品策略、成本控制策略等协同。**开源软件风险大小与产品商业模式息息相关，企业需要根据产品商业模式选择合适的开源软件。如果是在开源软件基础上进行二次开发形成闭源软件出售，就要避免使用强著佐权型许可协议的开源软件。**二是，开源法律风险防控仅仅单个部门很难独立完成，需要法务、技术、采购等部门协同配合。**例如，开源软件法律风险判断时，需要法务人员解读相关许可协议、技术人员分析软件的使用方式，共同判断风险大小；当采购的软件中包含源代

码时，还需要采购部门在采购合同谈判的环节与供应商协商好知识产权担保问题。企业可成立开源委员会或开源工作小组等机构，多部门人员共同参与，协商处理开源法律风险相关事宜。

（二）建立开源软件风险防控全流程管理

1. 开源软件引入时，综合考虑开源软件风险

企业引入开源软件时，要做好开源代码的审查工作，可以从源头减少开源带来的知识产权风险。一是，要确定开源代码所遵循的开源许可协议，判断该开源许可协议是否和企业产品的盈利模式相冲突。二是，了解开源社区的知识产权规定，判断开源代码的来源是否可靠，开源社区的知识产权规定是否对公司产品产生影响。三是，结合企业对开源软件的不同使用场景，综合判断开源软件的知识产权风险。

2. 软件开发过程中，做好开源代码跟踪管理

针对在开源代码基础上进行二次开发的软件项目，尽量采用模块形式进行链接，而不是将开源代码直接加入程序。技术部门需对软件开发使用的所有各种开源代码进行记录，包括开源代码模块的名称及版本，使用许可协议类型等，便于长期跟踪管理开源代码的使用情况。除此之外，企业还要定期跟踪开源许可协议的变化。企业收并购、产品业务调整、竞争策略变化等因素会导致开源企业更改开源软件的许可协议，甚至将开源产品变为闭源产品，这些变化往往会对软件的后续使用产生重大影响，因此需要及时获知许可协议变更情况，并重新评估使用风险。

3. 开源产品发布前，做好产品审核

在推出开源相关的产品时，需要法务人员做好产品审核。审核具体内容包 括：开源模块是否遗漏、是否按规定进行开源相关声明、获取源代码方式是否可用等。

（三）法务部门需做好开源相关法律研究工作

1. 准确解读开源许可协议规则

对开源软件法律风险影响最大的因素就是开源许可协议，因此法务部门需要准确解读企业常用的开源许可协议。不同开源许可协议对于版权、专利、商标等方面的规定有很大不同。解读许可协议时，需要重点关注的内 容包括：许可协议对术语的定义、软件再发布的规定、是否有专利许可、是否有专利报复、是否有不担保条款、是否规定法律纠纷管辖地等。

2. 跟踪开源相关司法案例

随着开源业务在新兴领域发展越来越重要，开源的知识产权纠纷必然会越来越多。法务人员需要对已经发生的开源纠纷案例进行跟踪分析。通过对案件争议焦点、判决结果的分析，一方面了解法官对开源软件知识产权问题的判决依据，为以后开源法律纠纷应对打好基础，另一方面掌握开源软件的诉讼动向，便于公司提前做好诉讼防控准备。

附件：开源知识产权热点事件分析

近几年，随着开源产业不断扩大，企业成为开源的主导力量，开源厂商与闭源厂商、开源厂商和二次开发者之间的利益冲突越来越严重，导致开源知识产权纠纷也愈加频繁。本研究报告梳理了近几年发生的开源知识产权热点事件，并对事件进行解读，供企业参考。

（一）ARTIFEX 诉 HANCOM

1. 事件基本情况

2016 年底，Artifex 公司因 Hancom 违反开源许可协议向美国加利福尼亚北部地区法院提起诉讼。被告 Hancom 公司是一家韩国企业，其出售的 Office 软件程序中，下载并使用了 Artife 公司所开发的 Ghostscript 工具。Artifex 公司是 Ghostscript 的开发者，Ghostscript 遵循 AGPL、商业双重许可协议，使用者可以遵循 AGPL 许可协议免费使用并发布对 Ghost 所做的任何修改；或者选择向 Artifex 公司支付许可费，并保留源代码。

Hancom 违反了 AGPL 许可协议，在集成 Ghostscript 到公司产品中时，没有按照 AGPL 许可协议所要求提供 Ghostscript 的修改代码。Artifex 要求 Hancom 补缴许可费，认为对方通过出售侵权的软件而获利数亿美元。Hancom 回绝之后，Artifex 将 Hancom 起诉至法院，要追溯许可协议授权费用。

2. 事件解读

该案件的争议焦点在于“违反 AGPL 开源许可协议是否是构成版

权侵权”。Hanco 公司辩护称：由于自己当初下载 Artifex 的软件时没有签名任何东西，所以不用执行任何合约。法官认为：如果用户没有获得商业许可协议，即可视 Ghostscript 的用户同意 AGPL 许可协议条款。人们在下载软件的时候默认同意 AGPL——这意味着 AGPL 本身就是把这个简单的版权问题并进入合同法领域所需要的“额外元素”。

从这个案例来看，GPL 类许可协议是自我传播的，即使没有“签署合同”，当你选择开源路线时，它就被认定为具备强制执行力的合同。该案例树立了像 GPL 这样的开源许可协议可以被视作法律合约的先例，在开发商违反了开源许可协议时会被起诉。因为美国属于判例法系，所以这个案件，就变成了开源软件在法律领域新的里程碑。

（二）Facebook 修改开源许可协议

1. 事件基本情况

2013 年，Facebook 公司开源 React.js，使用 BSD 许可协议。许多著名公司都采用该框架开发产品，包括 Microsoft、Airbnb、百度、阿里巴巴、腾讯、美团、携程、去哪儿，知乎等，使用非常广泛。2016 年 7 月，Facebook 在原有的 React.js 开源许可协议中附加了“专利反击”。2016 年 11 月，Facebook 发布官方问答，对附加专利条款进行澄清。2017 年 7 月，Apache 基金会把 Facebook BSD+Patents 加入了禁止名单中。随后，Wordpress、百度等多家企业宣布抵制 React。2017 年 9 月，Facebook 宣布将部分开源项目（包括 React、Jest、Flow

和 Immutable.js) 协议更改为 MIT；但是，Facebook 仍有一些项目（如 React Native 等）使用“Facebook BSD+Patent”的许可协议。

2. 事件解读

“Facebook BSD+Patents”许可协议是由标准的 3-Clause BSD 协议和一份专利授权组成。3-Clause BSD 许可协议是非常宽松的许可协议，商业友好，无专利规定。Facebook 附加的专利授权协议则存在“Patent retaliation clause”（专利反击条款），具体规定如下：

如果您（或您的任何子公司、分公司或代理商）直接或间接发起任何专利主张，或者从专利主张中获得直接的经济利益，那么本协议授予您的许可将自动终止。专利主张包括：

(i) 针对 Facebook 或其任何子公司或者关联公司；

(ii) 针对任何实体的专利主张，如果该专利主张全部或部分来自于 Facebook 或其任何子公司或关联公司的任何软件、技术、产品或服务；

(iii) 针对于该软件相关的任何一方。

尽管有上述规定，如果 Facebook 或其任何子公司或关联公司首先向您提起专利侵权诉讼，并且您在该诉讼中针对于该软件无关的诉讼提出了专利侵权反诉，根据本许可协议授权的许可不会由于这种反诉而根据本款第 (i) 款终止。

从上述规定可以看出：该专利许可协议的目的是防范使用 Facebook 开源软件的公司向 Facebook 提起专利诉讼。但是，该协议

之所以广受业界诟病，主要是因为这是一个单边优惠协议。只要某家企业使用了以“Facebook BSD+Patents”许可下的任一软件，那么就不允许向 Facebook 或其任何子公司或者关联公司发起诉讼，也不可以针对该软件的任何相关方提起诉讼。

使用 Facebook 开源软件的企业是否接受该协议，取决于各个企业对知识产权风险的判断以及软件替换成本之间的权衡。但是，此事也给企业使用开源软件敲响了警钟。使用开源软件虽然是“免费”的，但是背后可能也要付出“昂贵”的代价。

（三）MongoDB 变更开源许可协议

1. 事件基本情况

MongoDB 采用商业许可、开源许可双许可协议。MongoDB 商业版本采用商业许可协议，MongoDB 社区服务器版本之前采用 AGPL v3 许可协议。2018 年 10 月，MongoDB 宣布其开源许可协议从 AGPL v3 切换到 Server Side Public License (SSPL)。2018 年 10 月 16 日当天或者是以后发布的所有的 MongoDB 社区服务器补丁和新版本都采用 SSPL 许可协议，包括旧版本的未来的补丁。

SSPL 许可协议是在 GPL v3 基础上修改得到的。GPL v3、AGPL v3、SSPL 这三个许可协议的差异主要体现在第 13 条款。按照 GPL 许可协议的规定，任何人都可以以提供技术服务为目的，运行私有修改的 GPL 许可下的程序，只要不发布软件，就不需要公开源代码。但是，AGPL 许可协议将 Copyleft 这一概念延伸至网络上所交付的软件服务。

在 AGPL 的规定中，如果其许可的程序与用户通过网络进行远程交互，那么就需要提供源代码给用户，所有的修改也需要提供给用户。常见的“通过计算机远程网络交互”的场景有：网络和邮件服务器、基于互动的网络应用程序和在线播放的游戏服务器等。在 SSPL 许可协议中，明确规定将程序或程序的修改版本的功能作为服务向第三方提供时，需要提供“服务源代码”。最为典型的场景即云平台提供商将软件托管产品打包成服务。

2. 事件解读

MangoDB 之所以更换其许可协议，主要是因为：**一是，AGPL v3 许可协议的规定不明确，导致很多企业一直在试探 AGPL 的边界。**AGPL v3 第 13 条款规定了“远程网络交互”的限制条件。但是，业界对于 AGPL 远程网络交互的触发条件以及范围存有争议。因此，SSPL 明确规定了将程序作为服务提供的条件限制。**二是，IT 产业逐渐向服务化转型，MangoDB 需要更改许可协议以避免其商业模式受到冲击。**MangoDB 采用 AGPL 许可协议时，一些云服务厂商将 MangoDB 的社区版本修改后向用户提供其数据库的托管商业版本，而不将修改的源代码公开回馈给社区。如此一来，这些云服务厂商相当于从 MangoDB 企业版销售中分了一杯羹，抢占了其销售份额。MangoDB 更换许可协议就是要遏制云服务提供商攫取开源软件价值却不给予开源社区任何回报的行为。

许可协议更改对 MangoDB 社区服务器的常规用户影响不大，主要是对云服务商产生较大影响。MangoDB 更换许可协议之后，云服务厂

商如果想销售基于 MangoDB 的云服务，要么需要完全公开其服务源代码，要么需要向 MangoDB 购买商业许可。MangoDB 许可协议中，对“服务源代码”的范围界定非常宽泛，不仅包括 MangoDB 或其修改版本对应的源代码，还包括管理软件、用户界面、应用程序接口、自动化软件、监控软件、备份软件、存储软件和托管软件的源代码。将这些“服务源代码”完全公开，意味着这些云服务厂商丧失了产品差异化的能力。因此，其他云服务厂商可能也就没有积极性销售基于 MangoDB 的云服务。

CAICT 中国信通院

CAICT 中国信通院

中国信息通信研究院

地址：北京市海淀区花园北路 52 号

邮政编码：100191

联系电话：010-62304212

传真：010-62304980

网址：www.caict.ac.cn

