

“开源软件”风险防范

作者：杨迅

“开源软件”是免费软件吗？

“开源软件”可以自由使用吗？

“开源软件”在商业实践中应当防范哪些风险？

前言

开源软件不仅是软件爱好者的专属，其在现代软件行业中更有着广泛的商业应用。无论是为了节省软件开发的成本，专注于开发核心系统解决方案，还是为了促进系统生态的形成，很多软件公司都或多或少地在开发活动中使用开源软件代码。手机 **Android** 系统是基于开源软件开发的商业系统的成功范例之一。但是，使用开源软件并不能随心所欲，如果无视开源软件的许可规则任意使用，则可能给整个软件开发活动带来巨大的风险。

本文将简要介绍开源软件的法律性质、主要的开源许可证，以及使用开源软件的风险和防范。

第一部分 开源软件的前世今生

计算机软件的发展经历了一个从开源到闭源，再到开源的过程。上世纪六七十年代软件产业诞生之初，大部分的软件都以开源形式发布。这些软件并不是独立的产品，其往往附随着硬件设备一起被出售和使用，就像今天我们买的自动洗衣机、电视机中预先装载的程序一样，主要作用是实现硬件设备的功能。这段时期的软件，相对而言功能比较单一，且兼容性较差，所以无法在不同的设备间实现通用。因此软件产品仅是硬件设备的附属品，而以开源形式提供软件有助于在硬件设备或软件系统出错时进行及时的编辑、更改、和维修。

.....
如您需要了解我们的出版物，
请联系：

Publication@linkslaw.com

到了上世纪七十年代，随着计算机系统和编译器的发展，制造软件的成本，相对于硬件也日益增高。同时，软件的功能日趋丰富，已经能够实现在一件设备上运行不同功能的软件产品。这时，仅将软件作为硬件的附属品出售就限制了用户的选择，故而促成了软件逐渐被作为独立商品出售。

软件公司基于对自己产品的保护，避免竞争者抄袭自己的软件产品，开始通过编译手段将软件的目标代码出售，而源代码以商业秘密的形式保护起来，此时渐渐形成了以闭源形式出售软件的商业模式。美国国会在 1974 年成立了国家新科技应用版权作品委员会(Commission on New Technological Uses of Copyrighted Works)，认定计算机程序同样属于一种个人创作的类似文字作品，应当在版权法下获得保护，这开创了软件知识产权保护的先河。

于是，软件产生了独占的商品属性。软件产品再也不能被任意的复制、使用和传播，而软件的源代码则作为软件公司的核心资产被严密的保护起来。

而这种对于源代码的保护，实际上很大程度限制了软件爱好者学习、分享和改进现有的软件，也在很大程度上限定了软件的兼容性应用。1976 年比尔盖茨还写了一篇文章，《致爱好者的公开信》(An Open Letter to Hobbyists)，文中盖茨说，业余爱好者群体没有支付许可费用就大量分享微软产品 Altair BASIC(一种 BASIC 语言的编译器)，他认为业余爱好者没有支付许可费用就大量分享微软产品，对开发者付出的时间、精力是不公平的，并认为这是一种盗版的行。

但并非所有人都认可这种观点。美国著名的程序员、自由软件活动家，大名鼎鼎的理查德·斯托曼(Richard Stallman)担心软件爱好者不能学习、修改他人现有的软件，他认为软件的开发是一种思想的交流活动，公司将源代码保护起来阻止源代码的自由共享是一种伦理上的错误。1983 年，他在麻省理工学院发起了著名的 GNU 计划，针对当时被垄断的但却具有普遍价值的 Unix，建立自由软件集体协作(Free software foundation)，目的是开发出一套完全自由(Free Software)的操作系统。因此才有了与 Unix(当时最流行的闭源操作系统)兼容的操作系统，并将其命名为 GNU(GNU's not Unix)。他还发布了《GNU 宣言》，解释发起这个计划的初衷，理由之一就是“重现当年软件界合作互助的团结精神”。自此诞生了开源版的操作系统，从而实现了自由软件的回归。

第二部分 开源软件的法律性质

开源软件顾名思义是将源代码向公众公开的软件，其目的是允许不特定的第三方自由地使用、修改或创作衍生作品，也就是进行迭代开发，但开源并不抵制商业收费。

开源软件不代表该软件不受著作权保护。计算机软件享有的著作权，是在软件一经编写完成就自动获得的，无论是否开源都受到著作权法的保护。具体而言，软件的表达方式包括编码、流程设计，都可能受到著作权法保护，其背后的商业方法如果具有新颖性和非显而易见性，也可能受到专利法的保护。不仅开源软件受到著作权法和其他知识产权法的保护；并且，开源软件著作权人，也是积极地运用了著作权法和其他知识产权法的机制，规范其软件的使用和被利用。

开源软件并非孤儿作品。所谓孤儿作品一般指权利人不明或权利人无法联系的著作权作品。开源软件虽然为公众使用,但一般有明确著作权人(譬如为自由软件基金会或 NGO 所有),并且由著作权人行使著作权或其他知识产权相关权利。比如,当开源软件使用者违反许可证时,著作权人或其他权利人可以依据许可证,向其提出违约或侵权主张。

自由软件是一项思想运动,强调用户拥有如何使用软件的自由。所谓自由,强调的是使用软件对所有的用户来说是自由的。反对软件私有,首先反对的就是软件通过著作权和专利权等知识产权手段的私有化。为了表达对 Copyright(知识产权)的憎恶,理查德·斯托曼(Richard Stallman)甚至生造了一个单词 Copyleft(翻译为“著佐权”)。自由软件的著作权人,拥有其开发的开源软件的著作权和其他有关知识产权,并且通过向公众发放许可证的方式,允许不特定的第三方在满足许可证的条件下进行使用,但往往(但不必然)不允许第三方再将软件私有化。

开源软件和自由软件这两个术语通常具有相近意思。一般而言,开源软件强调其源代码的开源,即依照开源许可证的条款向公众公开。而自由软件强调的是自由使用、修改、发行以及自由创设衍生版本的权利,但也必须允许不特定的第三人也能自由地运行、拷贝、修改和再发行该软件以及在该软件的基础上加以修改而形成的衍生软件产品。开源软件的目的,在于其自由的使用、修改、创设衍生版本,自由使用、自由修改、自由发行、以及自由创设衍生版本依赖于源代码的开源,侧重点不同,但指的是同一软件。开源软件既继承了自由软件所提倡的知识共享的理念,同时又允许人们以专利的形式从知识产品中谋取利益,从而保护了人们生产、创造知识产品的积极性。

开源软件不等同于免费软件,免费软件就是免费提供给用户使用的软件。虽然自由软件和免费软件的英文表述都是 free software,但是含义却是不同的。免费软件不一定是自由软件,而且一般不是自由软件或开源软件,免费软件只不过表明其使用不向使用者或公众收取费用,但是通常都会有其他的限制,比如其源代码不一定会公开,而且使用者也并没有复制、研究、修改和再发布的权利。开源软件或自由软件也不一定是免费的,虽然往往是免费的,但自由软件仍然可以通过发行版本提供服务收取费用。

开源软件也不一定不是商用软件,通常开源软件的发行受制于一定的开源条件。但并不否认可以用在商业环境中,比如安卓系统是成功的商业软件,同时也是商业软件的典范。开源软件也不必然可以不受限制地用于非商业环境比如学习研究创作等活动。虽然开源软件的诞生宗旨是为了交流思想、促进研发,但无论是生产,还是研究学习环境,都必须遵守开源许可证的要求。

总而言之,开源软件是一种受到著作权保护的,并且利用著作权、知识产权机制,保证其自由修改发行使用和创作衍生作品的软件,因而使用开源软件必须遵守该开源软件所对应的开源软件许可证。

第三部分 开源许可证

开源软件是依照开源许可证发布的软件,其使用、发布、修改和创作衍生作品,也应当遵守开源许可证(License)的要求。如果在使用开源软件时没有遵守其许可要求的各项义务,就是违规使用开源软件,会造成许可证的合规性风险。

开源许可证的法律地位在美国法律下仍有争议，但在中国法律下，一般都被认可是一种软件所有人和使用人之间的许可协议。如果违反该协议条款使用开源软件的，则可能面临违约诉讼，造成公司产品的召回、被迫开源自有知识产权代码等等法律后果，而这些事件还可能对公司的声誉、市场准入等造成长期的负面影响。

开源许可证通常规定：如果违反条款使用软件的，会导致撤销，所以违反开源许可证使用开源代码的行为可能构成侵权。因此在使用开源软件时，留意许可证条款是防范风险的保证，也是合法合规使用开源软件的重要前提。

开源许可证的种类已经无法统计，理论上讲，每一个软件的作者都可以发布其自己的开源许可证，作为允许他人使用开源软件的条件。开放源代码促进会(Open Source Initiative)是一家位于加州的非盈利组织，其认可收录的开源许可证就有 110 余种。

开源许可证的条文通常比较难懂。这是因为：第一，很多开源许可证由计算机工作者编纂，因此很多语言习惯和正式法律文件不一样；第二，很多开源许可证的编纂往往针对特定技术问题，因此需要一定的技术背景才能理解；第三，开源许可证希望在各国法律下都能适用且具有相同含义，因此极力避免使用在国家法下具有特别含义的用语。总的来说，开源许可证分为两大类：一类是通常与自由软件运动相关的许可证，其被称为“copyleft” 软件许可证(下文将详细介绍)，其开源要求严格，相对于著作权 copyright 而言，更强调使用开源许可证下的代码进行修改、迭代或二次开发所得的软件作品向第三方开源。另一类是允许性许可证(permissive license)，其条款相对宽松，允许第三方使用开源代码进行编纂、二次开发所得软件进行商用，而不必向第三方开源。

最常见许可证有以下几种：

(一) GPL。最典型的著作权式(Copyleft)的开源许可证是 GPL (General public license)。Linux 就是采用 GPL 发布的。GPL 的突出特点是代码的开源，即被许可人可以免费使用、修改、创作衍生作品，但不允许修改后和衍生的代码作为闭源的商业软件发布和销售。也就是说，如果任何人采用了 GPL 保护的软件代码用于其自己软件作品，其作品也必须为开源的。从 GPL 条款来看，其对第三方的许可应当符合两个条件：其一是下游的自动许可，当使用人提供一份受保护的作品时，接受到这份软件作品的人，自动获得一份在 GPL 许可证下使用、修改、发布该软件作品的权利。其二，开源的传递性，要求使用 GPL 创作的开发者，当他提供一份基于 GPL 保护下开源代码的作品时，发布应当以源代码的形式进行；如果发行目标代码的，也应当同时提供相应源代码。换言之，使用 GPL 许可证下代码创作的作品，也应当是开源的。这里受保护的作品范围很广，包括在 GPL 保护的代码基础上的开发，也包括使用 GPL 保护下代码的软件产品。唯一可以不开源的例外是，不提供作品，比如仅仅以 GPL 保护下的作品提供服务，但并不把这份作品提供给第三方，这样可以避免开源的要求。

(二) AGPL。第二种常见的开源许可证 AGPL(Affero General Public License)。AGPL 是 GPL 的补充，是比 GPL 限制更加严格的许可证。AGPL 这个协议的制定是为了避免一个 GPL 或 LGPL 协议中的漏洞。GPL 针对传统软件的分发模式 (以微软为代表)，而 AGPL 的产生是针对于日渐广泛的软件服务化(SaaS)应用，

其目的是确保对开放源代码 SaaS 应用代码的修改会反馈给自由软件社区。在该种情况下, 软件并不以版本分发的形式商用, 而是在云空间以服务的方式提供给公众使用, AGPL 许可证要求, 哪怕以云空间服务形式提供给第三方或公众, 如果使用了 AGPL 仍然需要向公众开源。

(三) LGPL。LGPL(Lesser GPL)相对 GPL 来说开源较为温和, 可以通过引用类库的方式(不是直接使用源代码), 根据 LGPL 授权开发商业软件。简言之, LGPL 指给以链接方式调用类库的作品提供开源要求的豁免。该豁免开源要求的条件是: 一项需要使用 LGPL 许可证下的数据库, 如果能够保持该数据库的独立性, 则软件整体是允许闭源的。从技术角度来说, 如果一件软件作品以动态链接的方式, 调用一个 LGPL 许可证下的数据库, 该软件可以是闭源的且不必受到 LGPL 限制的。但如果 LGPL 许可证下的数据库, 是以内嵌方式, 嵌入一项软件作品, 那么整个作品必须根据 LGPL 规则开源、允许不特定的第三方自由使用。

(四) MPL。MPL(Mozilla Public License)与 LGPL 类似, 介于 GPL 和 LGPL 之间。LGPL 针对的是数据库, MPL 针对的是一件软件作品。如果一个软件是相对独立的, 只是调用 MPL 许可证下软件的某些输出, 则相对独立的这个软件可以是闭源的, 不受 MPL 许可证限制。相反, 如果一个软件受 MPL 许可, 以内嵌方式嵌入另一个软件, 那么整个作品必须根据 MPL 规则向任何第三方开源, 允许任何第三方自由使用。

(五) BSD、MIT 和 AL。在允许性许可证方面, 常见的一种是 BSD(Berkeley Software Distribution), AL(Apache License), MIT(Massachusetts Institute of Technology)。AL 允许公众和第三方使用 AL 许可证下的软件, 包括修改和再发布, 条件是: 第一, 需要给收到代码的用户一份 AL 许可证; 第二如果修改了开源代码, 需要在修改文件中作出说明; 第三, 如果根据 AL 许可证下的软件代码创作了新的作品, 需要带有源代码的协议声明和原作者需要包含的文件; 第四, 不能够使用原作者的名称、商标、服务标识等, 为修改后的作品或演绎作品做推广。可见, AL 许可证对受保护的的使用人没有继续开源的要求, 使用 AL 许可证保护下的代码或以此为基础创作衍生作品的, 可以做商业性的使用不必开源, 条件要尊重原作者的人身权利。BSD 许可证和 MIT 许可证则更为简单, 一般仅要求标注著作权标示, 或按照要求发布版权声明, 而没有其他使用的条件。使用 MIT 或 BSD 许可证下的开源代码而创作的作品不必向第三方开源, 可以商用。

第四部分, 开源软件的法律风险管理

通常来说, 软件程序主要来源于自主研发、购买产品、使用开源软件或采用软件外包开发等途径。据 VeraCode 公司统计, 大约 70%自主研发软件的代码都使用过第三方代码, 并多以开源组件、商业或外包共享库的形式存在。现代软件大多数是被“组装”出来的, 不是被“开发”出来的。

尽管使用开源软件可以很大程度上节约研发成本, 能够把主要资源投入到核心的开发工作上去。但是在企业使用开源软件时, 即视为已经接受了开源协议。如果技术人员在使用开源软件为企业进行软件研发时, 无视开源软件所附带的开源许可证的类型和要求, 随意地使用开源代码, 编写软件产品, 那么企业最终的软件研发产品, 可能会因为违反开源许可证给相关软件业务带来极大的风险。

首先, 这种风险可能就是开源风险, 比如使用一些具有开源传染性的许可证下的开源代码, 如 GPL 许可证下的开源代码, 可能导致整个软件产品被要求向公众或第三方开源, 允许第三方自由使用, 比如说华硕公司曾经被 Linux 社区成员指控使用 GPL 许可证(EeePC)编码而没有遵守其开源要求, 最终被迫将其产品开源。

其次, 未能遵守许可证要求还可能导致违约或者侵权诉讼, 例如, 在数字天堂北京网络技术有限公司诉柚子北京科技有限公司, 以及不乱买电子商务北京有限公司与北京闪亮信息技术有限公司的案件中, 被告由于未遵守开源软件许可证, 将具有开源要求的软件产品作为一般商业产品, 进行商业化许可第三方使用收取许可费用, 被开源软件的著作权人诉请违约和侵权, 故而承担法律责任。

可见, 如果不恰当地使用开源软件, 可能给软件业务带来不可测的法律风险。对于很多软件开发者而言, 使用开源社区的代码已经成为一种工作习惯, 很大程度上这也是软件业的一个特点。

那么, 在这种情况下, 如何防范使用开源软件的风险呢?

我们不能指望每一个软件开发者都熟悉开源软件的法律问题, 毕竟术业有专攻。因此, 这就需要企业建立一套关于开源软件的管理制度, 将对开源管理的使用纳入研发活动的日常管理中去。其一, 当软件开发者使用开源软件时, 应当如何报告、如何评判、如何确定使用开源软件的条件, 以及作出是否使用、如何使用的决策。其二, 企业应当建立一个机制, 关注和确定、了解开源许可证的要求。当必要使用开源代码时, 通过管理手段要求软件开发者, 针对开源许可证的要求作出相应的设计安排, 比如, 对于 LGPL、MPL 下代码的使用, 必须以动态链接的方式执行, 要避免内嵌形式。同时企业应当监控, 避免使用没有开源许可证的代码, 或开源许可不明确的开源代码, 避免不可测的风险。其三, 企业在研发活动中间, 应当对开源代码的使用进行长效的监测, 避免由于改变软件结构、设计安排或者软件的商业执行方式, 而导致违反开源许可证或落入强制开源要求的范围中。

如您希望就相关问题进一步交流, 请联系:



杨 迅
+86 21 3135 8799
xun.yang@llinkslaw.com

如您希望就其他问题进一步交流或有其他业务咨询需求, 请随时与我们联系: master@llinkslaw.com

上海

上海市银城中路 68 号
时代金融中心 19 楼
T: +86 21 3135 8666
F: +86 21 3135 8600

北京

北京市朝阳区光华东里 8 号
中海广场中楼 30 层
T: +86 10 5081 3888
F: +86 10 5081 3866

深圳

深圳市南山区科苑南路 2666 号
中国华润大厦 18 楼
T: +86 755 3391 7666
F: +86 755 3391 7668

香港

香港中环遮打道 18 号
历山大厦 32 楼 3201 室
T: +852 2592 1978
F: +852 2868 0883

伦敦

1/F, 3 More London Riverside
London SE1 2RE
T: +44 (0)20 3283 4337
D: +44 (0)20 3283 4323



www.llinkslaw.com



Wechat: Llinkslaw

本土化资源 国际化视野

免责声明:

本出版物仅供一般性参考, 并无意提供任何法律或其他建议。我们明示不对任何依赖本出版物的任何内容而采取或不采取行动所导致的后果承担责任。我们保留所有对本出版物的权利。

© 本篇文章独家授权威科先行法律信息库发布, 未经许可, 不得转载。