软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2022,33(1):1−25 [doi: 10.13328/j.cnki.jos.006279]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

开源许可证的选择: 挑战和影响因素*

吴 欣[1,2], 武健宇[1,2], 周明辉[1,2], 王志强[3], 杨丽蕴[4]

[1](高可信软件技术教育部重点实验室 (北京大学), 北京 100871)

[2](北京大学 计算机学院, 北京 100871)

[3](西南大学 计算机与信息科学学院/软件学院, 重庆 400715)

[4](中国电子技术标准化研究院, 北京 100010)

通信作者: 周明辉, E-mail: zhmh@pku.edu.cn

摘 要: 开发者通常会为其开源代码选择不同的开源许可证来约束其使用条件, 以期能有效地保护知识产权和维持软件的长远发展. 然而, 现有的开源许可证种类繁杂, 开发者难以了解不同开源许可证间的差异, 并且难以通过现有的开源许可证选择工具做出合适的选择——其使用要求开发者了解开源许可证相关条款并明确自己的业务需求. 学术界虽然对开源许可证已有研究, 但是对开发者选择开源许可证的实际困难并无系统的分析进而缺乏清晰的认知. 有鉴于此, 旨在从开源开发者角度出发, 理解其选择开源许可证的困难, 并通过分析开源许可证的组成要素和影响开源许可证选择的因素, 为开源许可证的选择提供借鉴. 设计问卷并随机调研了参与 GitHub 开源项目的 200 名开发者. 通过对 53 个反馈结果采用主题分析, 发现开发者选择开源许可证通常面临条款内容太复杂和考虑因素不确定这两方面的困难. 通过分析 GitHub 上 3 346 168 个代码仓库中使用最广泛的 10 种开源许可证, 建立了包含 10 个维度的开源许可证框架. 通过借鉴计划行为理论, 从行为态度、主观规范和知觉行为控制 3 方面提出了影响许可证选择的 9 大要素, 通过开发者调研验证了它们的相关性, 并进一步通过拟合次序回归模型验证了项目特征与许可证选择的关系. 研究结果能加深开发者对开源许可证内容的理解, 为开发者结合自身需求选择合适的许可证提供决策支持, 并为实现基于用户需求的开源许可证选择工具提供借鉴.

关键词: 开源许可证; 开源许可证框架; 开源许可证选择; 开源许可证选择的影响因素

中图法分类号: TP311

中文引用格式: 吴欣, 武健宇, 周明辉, 王志强, 杨丽蕴. 开源许可证的选择: 挑战和影响因素. 软件学报, 2022, 33(1): 1–25. http://www.jos.org.cn/1000-9825/6279.htm

英文引用格式: Wu X, Wu JY, Zhou MH, Wang ZQ, Yang LY. Selection of Open Source License: Challenges and Influencing Factors. Ruan Jian Xue Bao/Journal of Software, 2022, 33(1): 1–25 (in Chinese). http://www.jos.org.cn/1000-9825/6279.htm

Selection of Open Source License: Challenges and Influencing Factors

WU Xin[1,2], WU Jian-Yu[1,2], ZHOU Ming-Hui[1,2], WANG Zhi-Qiang[3], YANG Li-Yun[4]

[1](Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

[2](School of Computer Science, Peking University, Beijing 100871, China)

[3](College of Computer and Information Science/College of Software, Southwest University, Chongqing 400715, China)

[4](China Electronics Standardization Institute, Beijing 100010, China)

Abstract: Developers usually select different open source licenses to restrain the conditions of using open source software, in order to protect intellectual property rights effectively and maintain the long-term development of the software. However, since the open source community has a wide variety of licenses available, developers generally find it difficult to understand the differences between different open source licenses. And existing selection tools of open source license require developers to understand the terms of the open source

* 基金项目: 国家重点研发计划 (2018YFB1004201); 国家自然科学基金 (61825201)
收稿时间: 2020-05-09; 修改时间: 2020-07-13, 2020-11-09; 采用时间: 2020-12-09; jos 在线出版时间: 2021-01-15

license and identify their business needs, which makes it harder for developers to make the right choice. Although there has been extensive research on open source license, there is still no systematic analysis on the actual difficulties of the developers to choose the open source license, thus lacking a clear understanding. For this reason, this study attempts to understand the difficulties faced by open source developers in choosing open source licenses, analyzes the components of open source license and the factors influencing open source license selection, and to provides references for developers to choose open source licenses. This study conducts a random survey of 200 developers that participated in the open source projects on GitHub through questionnaires. With a Thematic Synthesis on the 53 feedbacks, it is found that developers often face difficulties in the selection of open source licenses in terms of complexity of terms and unknown considerations. By analyzing the ten open source licenses most widely used in 3 346 168 repositories on GitHub, this study establishes a framework of open source licenses that contains 10 dimensions. Drawing on the Theory of Planned Behavior, Nine factors that affect license selection from three aspects are put forward: behavior attitude, subjective norm, and perceived behavior control. The relevance of those factors is verified by developer survey. Furthermore, the relationship between project characteristics and license selection is verified by fitting the order regression model. The results of research can deepen developers' understanding of the contents of open source licenses, provide decision support for developers to select appropriate licenses based on their own needs, and provide a reference for implementing open source license selection tools based on developers' needs.

Key words: open source license; open source license framework; open source license selection; influence factor of open source license selection

在过去 30 年里, 开源已经成为软件技术创新和软件产业发展的主要模式. 与传统开发模式相比, 开源开发展现出充分共享、自由协同、无偿贡献、用户创新、持续演化的新特征, 颠覆了诸多经典软件工程的基本假设和理论[1], 吸引着越来越多的开发者和企业. 目前开源软件已经占据很大的市场份额, 例如, Netcraft 关于 2020 年 1 月 Web server 的市场调研数据 (https://news.netcraft.com/archives/2020/01/21/january-2020-web-server-survey.html) 显示, 开源软件 Nginx 和 Apache 分别以 37.70%、23.98% 的市场份额位列前二, 而 Microsoft 次之, 占比为 14.03%. 然而, 开源软件的开发和使用也伴随着许多的风险, 其中最大的风险之一就是潜在的知识产权侵权责任[2]. 为了保护开源软件的版权和进一步发展自由开源软件 (free/libre and open source software), 开源许可证应运而生. 开源许可证规范了软件的使用、修改、重新发布、担保和归属, 开发者通常会为其项目选择合适的开源许可证以保证自己的知识产权被合理利用.

开源许可证的选择对项目的开发和演化以及软件的使用和商业化而言都至关重要. Linus Torvalds 曾提到开源许可证是 Linux 成功的决定性因素之一, 并明确表示对 GPL-2.0 的偏爱, 因为它强制其他开发者进行反馈, 有效地防止了碎片化[3]. 同时, 开源软件所使用的许可证类型得到了其开发者以及用户的大量关注, 例如, Redis 变更模块开源许可证, 从 AGPL 迁移到 Apache-2.0 与 Commons Clause 相结合的许可证, 对销售许可软件作了限制, 在当时引起了极大的争议[4]; Google 为了吸引更多的厂商参与, 使用宽松型许可证 Apache-2.0 对 AndroidSDK 进行开源, 尽管 Android 是一个基于 Linux 内核 (遵循 GPL-2.0) 的操作系统, 然而 Google 独立开发的用于硬件驱动和 App 接口的中间层 AndroidSDK 却未遵循 GPL 协议, 其知识产权仍被控制在 Google 手中[5]; 为了方便中文用户, 北京大学牵头开发了中英文双语木兰宽松许可证 Mulan PSL 并迅速获得广大中文开源社区的采纳.

为项目选择一个合适的开源许可证并不简单. 一方面, 目前开源社区存在大量不同的许可证, 仅 OSI 认证通过的开源许可证已有 80 多个, 尽管每个开源许可证背后的基本原理是相似的, 但它们之间存在相当大的异质性[6]. 开源许可证之间细微差异所体现的不同法律含义常常让开发者感到困惑, 例如, Apache-2.0 和 MulanPSL-2.0 (北京大学联合国内开源社区、开源生态圈产学研各界优势团队以及拥有丰富知识产权相关经验的众多律师, 在对现有主流开源协议全面分析的基础上, 共同起草、修订并发布了木兰宽松许可证第 2 版 MulanPSL-2.0, 并获 得了 OSI 认证. http://license.coscl.org.cn/MulanPSL2) 都授予专利权, 但对于授权主体的范围表述不同, 且 MulanPSL-2.0 不需要开发者对修改进行声明, 从而降低产生法律纠纷的风险也更完善地保护开发者的切身利益. 另一方面, 不同的开源许可证可能对志愿者参与软件开发的吸引程度不同, 例如, Colazo 等人通过对 SourceForge 托管项目数据的分析发现 copyleft 型开源许可证比非 copyleft 开源许可证吸引了更多的志愿开发者[7], 从而一定程度上证明开源许可证的类型影响着开源软件发展. 此外, 开源许可证的选择是项目负责人将项目开源时首要考虑的配置因素之一, 他们通常认为同一种许可证无法满足不同项目的需求[8], 例如, 国内领先的互联网的安全服务提供商 360 公司在开源中国 (https://www.oschina.net/) 发布的 31 个开源项目分别采用了 Apache-2.0、MIT、BSD 系列、GPL 系

列的开源许可证.

为了帮助开发者选择开源许可证, 业界已经实现了许多优秀的开源许可证选择工具. ChooseALicense (https://choosealicense.com/) 基于几种简单的项目应用场景为开发者提供推荐, 例如, 对于依赖社区开发的开发者推荐简单且限制少的许可证 (如 MIT 开源许可证), 而对于偏向于共享源码的开发者建议选择 GPL-3.0 许可证等. ChooseALicense 的特点是简单易理解, 然而其推荐选择的范围单一且局限, 也没有对开发者的复杂需求进行考虑. OSSWATCH 的 Licence Differentiator (http://oss-watch.ac.uk/apps/licdiff/) 和码云 (https://gitee.com/) 代码托管平台集成的许可证选择工具, 主要原理是基于开源许可证部分条款之间的差异, 根据开发者的偏好进行选择, 帮助开发者缩小许可证选择的范围, 但使用这类工具的前提是开发者必须了解所选择的开源许可证条款的含义. 然而, 大多数开源许可证中包含大量生涩难懂的法律术语, 开发者难以直接从条款内容分析对比, 例如, 对于很多商业开发者来说准确理解 GPL 的含义是一个常见挑战[9]. 同时, 开发者参与开源的实际需求往往各不相同, 例如, 根据其商业模式获取商业利益、提升其产品的知名度以获取广大用户基础、提高其在开源社区的声誉、获得社区技术支持等[10], 且这些诉求往往不是单一出现的. 在与华为等企业的开源专家的访谈中, 他们提到希望开源软件在具有良好的兼容性和广大用户的基础上能够避免陷入法律诉讼. 而上文提到的 Linux 内核和 Android 也正是基于不同的诉求而选择了不同的开源许可证. 因此, 仅仅通过对许可证条款的选择偏好或者基于简单的应用场景的推荐, 难以帮助开发者做出最佳决策.

鉴于开源许可证对开源开发的重要影响, 软件工程领域对其相关内容已有深入研究, 主要体现在对开发者选择或变更开源许可证的动机、以及开源许可证选择对项目成功的影响上. 例如, 开发者的经济动机决定开源许可证的选择[11], 许可证类型影响用户兴趣和贡献者的数量, 并与软件开发速度、开发者提交贡献的频率和参与的持久性相关[7,12]等. 这些研究一定程度上提高了开发者对开源许可证的认识. 然而, 随着开源的不断深入, 开发者选择开源许可证的困难愈发突出, 学术界对开发者选择开源许可证的实际困难并无系统的分析, 进而缺乏清晰的认知. 因此, 本文从理解开发者选择开源许可证所面临的具体困难出发, 通过分析许可证组成要素和影响开源许可证选择的因素, 来帮助开发者更好选择开源许可证. 具体来说, 本文主要回答下述 3 个研究问题:

RQ1: 开发者为项目选择开源许可证时通常会面临哪些困难?

RQ2: 开源许可证的组成要素有哪些?

RQ3: 哪些因素影响开发者选择开源许可证?

本文采用定性和定量相结合的方法回答上述问题, 首先通过阅读现有文献及结合有关项目开发经验, 我们设计了调查问卷, 从 GitHub 项目仓库的作者中选取 200 名开发者进行问卷调研, 并基于主题分析的方法总结出开发者选择开源许可证通常遇到的两类困难, 分别是: ① 开发者通常难以理解许可证的条款内容, 许可证之间的相似性及其复杂的法律含义让开发者感到困惑; ② 开发者选择开源许可证时通常受到多种因素影响, 例如项目的特征、开源许可证是否被广泛使用以及开源许可证对项目是否产生影响等, 他们对如何全面考虑各方面因素进行最佳决策感到困惑. 其次, 本文通过对 GitHub 上使用最广泛的 10 种开源许可证进行对比分析, 采用主题分析的方法提取了一个十维度的开源许可证框架, 可以帮助开发者认识开源许可证的组成要素和分析开源许可证间的差异. 再次, 我们借鉴了计划行为理论中的 3 个维度, 通过问卷调研以及相关文献调研, 将开发者选择开源许可证的考虑因素概括为 9 个方面, 包括: 个人的开源理念、对利益因素的评估、所在组织的观念、开源社区对许可证的偏好、许可证流行度和复杂度、许可证兼容性、其他项目影响、对项目特征的评估、以及许可证选择结果的影响等, 并进一步通过拟合次序回归模型验证了项目特征因素与许可证选择的关系. 最后本文讨论了研究结果的指导意义和应用场景.

本文的主要贡献总结如下:

① 调研并识别了开源许可证选择的两类常见困难: 开发者难以理解开源许可证的条款内容且许可证之间的相似性以及复杂的法律含义让开发者感到困惑; 开发者选择开源许可证时通常还会综合考虑多方面因素, 他们对如何全面考虑各方面因素进行最佳决策感到困惑. ② 围绕开源许可证核心要素建立了一个开源许可证框架, 可以帮助开发者方便理解开源许可证内容的构成以及开源许可证之间存在的差异. ③ 揭示了影响开发者选择开源许可证的 9 大因素, 可以指导开发者结合自身业务需求选择合适的许可证.

本文第 1 节对相关工作进行阐述. 第 2 节对开源许可证的背景进行介绍. 第 3 节探索开发者在选择开源许可证时通常遇到哪些困难. 第 4 节建立开源许可证框架, 以帮助开发者认识和了解开源许可证的构成和存在的差异. 第 5 节探索影响开发者选择开源许可证的影响因素. 第 6 节讨论了研究结果的指导意义和应用场景. 第 7 节阐述了本文的局限性. 最后进行总结.

1 相关工作

随着开源软件越来越广泛地被使用, 开源许可证也逐渐受到学术界和工业界更多的关注. 目前, 关于开源许可证的研究领域主要集中在法律[13], 经济管理[14], 社会学[15]以及软件工程等相关领域[2]; 在研究的内容上主要包括开源许可证的选择、开源许可证合规性使用[16,17]以及相关自动化工具, 如开源许可证选择工具[18]、开源许可证检测工具[19]、开源许可证管理工具[20]等方面. 而其中关于开源许可证的选择的研究主要集中在如下几个方向:

(1) 开源许可证的选择或变更的影响因素. 开源开发者的动机一直是与开源软件相关的研究人员和专业人士讨论的主题[21], Kaminski 和 Perry 认为许可证的选择主要取决于软件开发人员的意图和期望[2]; Lerner 和 Tirole 指出开源许可证的选择是由许可方和开发者社区的经济动机所决定的[11]; Singh 等人展示了开发者所处的社会环境如何影响他们对开源许可证的选择, 以及开发者的个人经历如何调节这种影响[15]; Sen 等人使用动机和态度理论来研究开发者对 3 种开源许可证类型的偏好[8]; Skidmore 讨论了不同类型的开源许可证的利益相关者的需求与义务[22]; Viseur 和 Robles 研究了开源项目中许可证变更的动机和影响[23]; Vendome 等人通过定量和定性的方法研究了 GitHub 上的 Java 项目中许可证何时以及为什么会改变[24].

(2) 开源许可证的选择或变更对项目的影响. Hofmann 等人分析了 10 年间开源项目的许可证选择和相关项目增长的趋势[25]; Stewart 等人调研了开源项目的开发活动以及用户兴趣与许可证类型的关系[12]; Colazo 和 Fang 基于社会运动理论研究了开源许可证的选择与项目活动之间的关系[7]; Kashima 等人进行了开源许可证对软件重用影响的定量研究[26]; Kechagia 等人探索了开源许可证之间的依赖关系, 用来解释和指导开源项目的许可证选择[16]; Jensen 和 Scacchi 通过对 Apache 基金会创建和迁移到 Apache-2.0 以及 NetBeans 项目迁移到 Joint Licensing Agreement 过程的案例研究, 分析许可证变更所带来的影响[27]; Wu 等人调研了由于许可证变更而导致包含不同许可证的相同原始文件的不一致性问题[28].

(3) 许可证选择自动化工具. 例如 Kapitsaki 和 Charalambous 采用相似用户和相似项目实现了开源许可证推荐[18].

此外, 还有大量的研究出现在经济法律和管理领域, 关注于常见开源许可证的对比分析[2,6,29−32], Valimaki[13]以及 Comino 等人[33]分析了开源公司如何使用双重许可, 确定了双重许可的法律和经济要求. 尽管这些研究提供了一些如何选择开源许可证的见解, 但它们通常仅在某一方面或个别细节上进行了深入的分析和建议, 缺乏对开发者实际需求的综合考虑和研究, 而选择开源许可证的影响因素可能是多方面的, 开发者难以直接将这些研究结论应用于不同的应用场景或变化的业务需求中. 同时, 业界对开发者选择开源许可证面临的实际困难还缺乏清晰的认识和系统的分析, 现有的开源许可证选择工具要求开发者清楚了解开源许可证相关条款的含义和明确自己的业务需求, 否则难以做出合适的推荐. 本文工作旨在从开发者角度出发, 发掘他们在选择开源许可证过程中遇到的实际困难, 提炼出具有一般性和通用性的开源许可证框架以及影响开发者选择的多个角度和因素, 为开发者选择开源许可证提供决策支持和经验参考, 同时为实现基于用户需求的开源许可证选择工具提供借鉴.

2 研究背景

2.1 开源许可证的产生

软件的版权保护意味着软件只能在得到版权所有者的许可后才能使用[2]. 因为软件很容易被复制, 但是创造它却是非常困难和代价高昂的[29], 所以通过版权法保护软件至关重要. 目前, 版权保护策略主要存在两种形式: 私有版权策略和非私有版权策略.

私有版权策略是一种将部分或全部潜在的技术用户排除在外的策略[34]. 在私有软件开发模型中, 代码首先受

到版权保护, 然后根据授权协议进行分发, 从而赋予用户特殊的权利[29]. 通过授权软件, 软件制造商可以限制用户的责任和权利, 例如: 只允许在一台计算机上使用等[31], 而用户需要为使用、分发、复制或编辑软件支付相应的版税.

非私有版权策略主要包括将软件置于公共领域或进行开源许可[34]. 将软件置于公共领域意味着完全放弃对其软件的版权保护, 任何人都可以无偿地使用和修改, 甚至可以删除作者的名字视为自己的作品. 软件的私有化被早期一些程序员认为是“不道德”的行为, 在 20 世纪 80 年代中期, 麻省理工学院的程序员 Richard Stallman 开发了一种新的软件分发方法, 即 GNU 公共许可证[35]. Stallman 关于自由软件的革命性思想随后演变为当前的开源软件运动, 自由/开源软件的主要目的是最大限度地开放以及减少软件使用传播创新的障碍[31]. 开源许可通过分配知识产权的权利来共享软件代码, 用户和开发人员社区可以自由访问, 以促进不同动机的参与者之间的合作和有益的交流[36].

综上所述, 开源软件仍受版权保护, 开源软件和私有软件的主要区别在于它们的许可模式. 使用开源软件需要得到开源许可证的授权.

2.2 开源许可证

开源促进会 (open source initiative, OSI)(https://opensource.org/) 创建于 1998 年, 是一个审查和批准开源许可证的非盈利组织. 他们为开源许可证建立了一套一致的标准, 称之为“开源定义”(open source definition, OSD)[11]. 同时, OSI 还注册了一个认证标志: OSI 认证标志 (the OSI logo), 这个标记可以放在发布的软件上, 这样人们就可以很容易地识别出这是开源软件并且所使用的许可证符合开源定义[6]. 通常我们认为符合开源定义的许可证就是开源许可证. 开源定义对于开源许可证的标准包含以下几个条件.

必须允许任何人以源代码或其他形式重新发布程序, 且不收取任何费用; 源代码免费可获取, 或收取不超过传输成本的费用; 必须允许分发衍生或修改后的软件; 不歧视任何用户群体或应用领域; 保证源代码的完整性; 允许对许可证中的权利重新分发; 许可证不得特定于产品; 许可证不得限制其他软件; 许可证必须是技术中立的 (https://opensource.org/osd).

2.3 开源许可证的类型

开源许可证间最大的差异是关于许可证对分发衍生软件的限制性不同, 即当他人对代码修改和扩展 (与其他软件合并) 后并分发的限制要求. 目前, 开源许可证按照限制的强弱通常分为 3 种类型.

● 宽松型 (permissive): 这类许可证通常只要求被许可方承认原始作者, 衍生软件可以成为私有软件, 如 BSD 系列许可证、MIT、Apache-2.0 和 MulanPSL-2.0.

● 限制型 (copyleft): 旨在促进开发人员的合作, 保护源代码的自由共享[35]. copyleft 条款要求对软件的修改和扩展, 必须按照获得该软件的许可证进行开源, 如 GPL 系列许可证、OSL 系列许可证.

● 弱限制型 (weak-copyleft): 弱限制型许可证要求对软件的修改, 重新分发必须按照获得该软件的许可证进行开源, 然而合并这些软件的大型作品可以成为私有作品. 这是一个折中的方法, 允许将代码集成到自己的软件中, 而不必使整个代码库开源, 避免了不得不分享的场景[37].

3 开发者选择开源许可证面临的困难 (RQ1)

我们在阅读大量的相关文献及与有关企业开发人员访谈后, 进一步通过问卷调查 (https://gitee.com/blesswoo/questionnaire/blob/master/关于开源许可证选择的调查问卷.md) 的方式, 来了解和分析开发者为项目选择开源许可证时通常面临哪些困难.

3.1 方法设计

(1) 问卷设计

我们通过阅读大量文献以及在线网页, 并咨询华为等企业开发人员, 初步分析了开发者选择开源许可证过程中可能遇到的困难. 以此为基础, 遵循相关性、完整性、互斥性和可能性的原则[38]设计了调研问卷. 针对调研对象, 我们设计了问题 (Q)1 开发者的所在地区 (填空)、问题 2 开发者参与开源年限考虑在内 (单选), 以期望问卷结

果具有广泛性; 针对开发者选择开源许可证面临的困难设计了问题 3; 同时, 我们还调研了开发者选择开源许可证的考虑因素设计问题 4–问题 11.

问题 3 共 7 个问卷选项 (多选, 详见表 1), 且包含开放式回答 (other) 的选项, 给予问卷回复的开发者表达自己的遇到的其他问题或者并没有在开源许可证选择方面遇到困难. 选项设计具体理由如下.

表 1 开发者选择开源许可证面临的困难

(问题3)在选择开源许可时遇到了什么困难?(多项选择题) (Q3)What difficulties have you met in choosing an open source license? (multiple-choice)		
选项	数量	比例 (%)
1. 开源许可证种类太多, 难以进行分析和比较(Too many open source licenses, and it's hard to analyze or compare).	21	39.62
2. 难以综合所有的因素来做出最佳的许可证选择策略(It is difficult to combine all the factors to make the best strategy for license choice).	8	15.09
3. 许可证中的术语或法律含义很难理解(Terms or legal implications of those licenses are difficult to understand).	21	39.62
4. 当不同项目之间存在依赖关系时, 很难判断许可证之间的兼容性(It is difficult to judge the compatibility of the license when there are dependencies among different projects).	15	28.30
5. 不同的许可证可能会对项目的发展带来不同的影响(Different license may have a different impact on the development of the project).	18	33.96
6. 不同的受众对许可证可能有不同的偏好, 比如贡献者或最终用户(Different audiences may have different preferences, such as contributors or end-users).	9	16.98
7. 现有的许可证无法满足自己的需求(Existing licenses cannot meet our needs).	0	0.00
8. 其他(Other).	11	20.75

首先, 很多网站都维护了一个开源许可证列表, 如 FSF (https://directory.fsf.org/wiki/Category:License)、OSI、SPDX (https://spdx.org/licenses/) 等, 开发者获得开源许可证的信息并不难, 难点在于开发者如何过滤这些大量信息以获得相关性, 以及利. 因此我们设计了选项 (option)1 和选项 2. 另一方面, 由于开源许可证的内容缺乏一致性和标准化, 例如, MIT 与 GPL 在内容结构上存在相当大的差异, 开发者并不总是清楚许可证中授权与限制的含义, 而许可证的法律性质加剧了这个问题准化, 例如, MIT 与 GPL 在内容结构上存在相当大的差异, 开发者并不总是清楚许可证中授权与限制的含义, 而许可证的法律性质加剧了这个问题[39], 因此设计了选项 3.

其次, 问卷中选项 4 的设计是因为在项目的开发过程中, 多个项目之间可能存在大量的交互, 如链接、合并、代码块复用等. 当所依赖的项目使用不同开源许可证时, 开发者不得不考虑许可证的兼容性及合规使用问题[16,17].

最后, 选项 5 和选项 6 的设计源于已有研究揭示了开源许可证类型与项目的开发活动、用户兴趣的关系[12], 以及开源许可证的选择对贡献者的动机和项目成功的影响[24]. 此外, 我们还发现除了 OSI 认证的 80 多个开源许可证外, 仍然有大量的不尽相同的开源许可证, 出现这种现象的可能原因是现有的许可证无法满足开发者的需求, 所以, 设计了选项 7 的选项.

我们从 GitHub 网站托管的代码仓库中利用 GitHub Search API (https://developer.github.com/v3/guides/) 收集了流行度排名前 10 的编程语言的共 9672 个项目仓库 (创建于 2018 年 1 月至 2019 年 9 月之间), 其中编程语言流行度参考了 TIOBE 编程语言 2019 年 9 月排行榜 (https://www.tiobe.com/tiobe-index/), 它反映了编程语言流行趋势以及某个编程语言的热门程度. 由于这 9672 个项目仓库存在重复的情况, 因此我们对数据进一步清洗, 通过 ID、创建时间的一致性去重, 共得到 7938 个项目仓库. 我们又从中剔除了许可证为空和标记为 other (通过人工随机抽查 5 个被标记为 other 的开源项目, 主要包括多重许可证、自定义开源许可证或正在实施许可证变更等情况) 的项目仓库, 最终得到了本次实验的 4704 个项目. 我们从 4704 名项目所有者 (owner) 中随机抽取 200 名 (参考了统计学关于简单随机抽样样本量计算方法[40], 按照调查结果在置信度为 95%, 误差范围在 4%–8% 之间的抽样样本数为 146–533, 同时为了避免产生过多打扰, 最终确定发放样本 200 份), 向他们发送邮件, 并附上利用问卷星

(https://www.wjx.cn/) 制作的问卷链接.

(2) 数据分析

我们使用主题分析方法对问卷结果进行分析. 主题分析 (thematic synthesis) 是一种识别、分析和报告数据中的模式 (主题) 的方法, 通常用于对定性研究数据进行分类, 在软件工程等众多领域广泛使用[41]. 主题分析一般分为 5 个步骤: ① 数据提取 (extract data), 是指从问卷答复中提取数据; ② 数据编码 (code data), 从数据集中标识和编码感兴趣的概念、类别、发现和结果; ③ 概念化 (translate codes into themes), 将标识和编码的内容总结为子主题; ④ 范畴化 (create a model of higher-order themes), 探索子主题间的关系并总结为更高阶的主题; ⑤ 验证 (assess the trustworthiness of the synthesis), 评估主题分析的解释的可信性.

步骤①通过对收到的回复, 提取出选项及补充的其他答案等信息; 步骤②从前面提取的信息中定位内容的关键点, 以系统的方式识别和编码有关开发者选择开源许可证过程中可能遇到的困难 (图 1); 步骤③通过对识别的开源许可证选择的多个困难对比, 总结共性和归纳主题形成概念性的主题; 步骤④对前面所归纳的主题再次抽象, 将相似问题归为一类, 在此基础上总结结论, 即开发者选择开源许可证面临的困难; 最后是步骤⑤, 本文的多个作者对分析的结果进行交叉验证, 获得了一致意见.

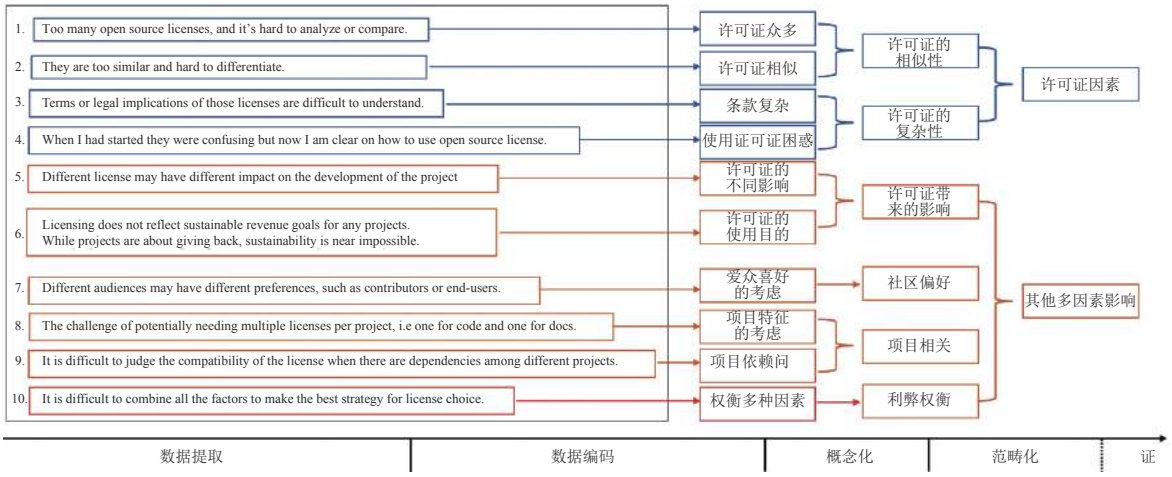


图 1 定性分析材料 (问卷选项及开放式回复) 主题分析过程

3.2 结果分析

我们向第 3.1 节选取的开发人员共发放 200 份问卷, 最终收到 53 份回复, 回复率为 25%. 通常, 软件工程领域中邮件调研回复率为 6%–36%[42], 因此我们的回复率在相对较高的范围. 其中, 回复问卷的开源开发者主要来自包括美国, 德国, 日本, 瑞典等 25 个国家, 样本的分布广泛. 这些开发者中 23% 参与开源 1–3 年, 77% 开发者有超过 4 年的开源经验. 84% 的开发者认为许可证的选择是困难的, 其中包含 83.33% 的开源经历为 1–3 年的开发者, 以及 79.31% 的超过 4 年以上开源经历的开发者. 仅 8 人 (15.38%) 表示没有困难, 其中 1 人提到“开始时, 开源许可证让我感到困惑, 但现在我清楚了如何去使用它们 (When I had started they were confusing but now I am clear on how to use open source license)”. 这些都进一步说明了开源许可证选择的困难是普遍实际存在的.

我们对得到的 53 份回复结果采用主题分析, 最终归纳出两类困难, 分别是:

① 开发者认为目前存在大量的开源许可证, 它们相互之间的相似性使其难以区分, 且一些开源许可证的条款和复杂的法律含义使其感到困惑. 39.62% 的开发者认为目前存在大量的开源许可证, 难以对比或分析; 而一些开源许可证的条款以及所包含的法律含义让人难以理解, 例如 Apache-2.0 和 GPL-2.0 中对专利授权的范围不同, 由于 Apache-2.0 包含了专利授权终止条款, 使得 Apache-2.0 无法兼容 GPL-2.0. 一名开发者在补充回答中提到“它们太相似了, 很难区分 (They are too similar and hard to differentiate)”. 开源许可证是相似的, 它们之间可能存在一些

共同的模式, 找到这种模式可能帮助开发者更好理解和对比开源许可证;

② 开发者选择开源许可证时可能综合考虑多种因素, 例如, 他们可能针对不同的项目或者项目特征为其选择适用的开源许可证, 开发者还关心许可证选择的结果是否影响项目的发展等, 开发者对如何全面考虑各方面因素进行最佳决策感到困惑. 33.96% 的开发者认为许可证选择的结果可能进一步影响项目的发展, 例如 MIT、BSD-2-Clause、Apache-2.0 等开源许可证在分发义务中允许衍生软件成为商业软件, 可能使得软件具有更多的用户, 这种影响的不确定性增加了他们选择开源许可证的难度; 28.3% 的开发者在选择开源许可证时容易受到许可证兼容性问题的困扰, 例如, 在 Linux kernel 基础上扩展软件需要遵从 GPL-2.0 的许可证约束, 而判断所选的开源许可证是否与其兼容需要一定的法律知识以及了解许可证的内容; 16.98% 的开发者认为, 针对不同的项目受众应当选择不同的许可证, 例如一些具有自由开源理念的贡献者通常会选择具有限制型许可 (如 GPL 类许可证) 的项目进行贡献, 一名来自美国, 超过 10 年的开源开发经验的人认为“每个项目可能需要多个许可证, 比如代码部分使用一个许可证, 文档部分使用另一个许可证 (The challenge of potentially needing multiple licenses per project, i.e. one for code and one for docs)”, 也进一步说明了开发者选择开源许可证式会考虑针对项目的不同类型选择不同的许可证; 还有 15.09% 的开发者认为影响开源许可证选择的因素是多方面的, 如何全面考虑这些因素进行选择是困难的.

此外, 有开发者表达了开源许可证具有一定的局限性, 尽管其规范了开源软件的使用、复制修改和分发, 但无法保证项目的可持续发展 (“Licensing does not reflect sustainable revenue goals for any projects. While projects are about giving back, sustainability is near impossible”). 值得注意的是, option7 结果为 0, 出现这个结果可能是因为, 我们选取的调研对象是从去掉许可证为空或其他许可证的项目作者中挑选, 而所调研的开发者认为目前的开源许可证已经能够满足他们的需求.

3.3 结　论

我们发现: ① 大部分的开发者在为项目选择许可证是困难的, 尤其是新参与的开发者在面对众多开源许可证, 它们内容的相似性和复杂的法律含义让人感到困惑. ② 开发者选择开源许可证时通常还会综合考虑多方面因素, 他们对如何全面考虑各方面因素进行最佳决策感到困惑.

4 开源许可证的组成要素 (RQ2)

为了帮助开发者更好地理解开源许可证, 以及减少开发者选择许可证的困难, 我们通过主题分析的方法探究开源许可证的组成要素.

4.1 方法设计

首先, 我们使用谷歌 Bigquery 工具 (https://console.cloud.google.com/) 获取了目前开源许可证的使用情况. 谷歌 Bigquery 是一种用于处理和分析大数据的 Web 服务, 其提供的 GitHub 的公共数据是迄今为止最大的 GitHub 可用数据源, 所收录的项目仓库均受到一个开源许可证约束. 截止至 2019 年 11 月 25 日, 我们分析了共 3 347 168 万个项目仓库, 提取每个项目授权的许可证信息, 并统计每种开源许可证的使用情况.

其次, 我们选取广泛使用的前 10 种许可证, 采用主题分析[41]的方法进行分析: 其中, 数据提取步骤中我们对选取的 10 种许可证, 提取出许可证的基本信息, 条款内容以及使用说明等; 数据编码的步骤中, 我们从前面提取的信息中定位内容的关键点, 以系统的方式识别和编码许可证的内容 (图 2); 概念化的步骤里, 通过对许可证内容的关键信息对比, 总结共性和归纳主题形成概念性的主题, 将许可证内容划分为多个维度; 范畴化的步骤里, 我们对前面所归纳的多个维度再次抽象, 总结出开源许可证框架.

最后, 为了检验本文提出的维度是否能反映大部分开源许可证的内容, 本文作者通过人工分析 OSI 或 FSF 认证的 72 个许可证内容对提出的维度进行交叉验证.

4.2 结果分析

我们对 3 346 168 个许可证信息进行统计, 得到图 3 的结果. 由图中可知 MIT, Apache-2.0, GPL-3.0, GPL-2.0,

BSD-3-Clause, BSD-2-Clause, AGPL-3.0, LGPL-3.0, CC0-1.0, EPL-1.0 这 10 种开源许可证是使用最广泛的, 占比 97%. 其中 MIT 许可证使用最多, 占比 51%; 其次 Apache-2.0 占比 15%; GPL-3.0 和 GPL-2.0 各占比 10%. 开源社区中普遍使用的开源许可证主要是 OSI 对许可证分类中的“流行且广泛使用的许可证”. 按照开源许可证的类型进行划分, 其中 MIT, Apache-2.0, BSD-3-Clause, BSD-2-Clause, CC0-1.0, ISC, Artistic-2.0 属于宽松型许可证, LGPL-3.0, EPL-1.0, LGPL-2.1 属于弱限制型许可证, 而 GPL-2.0, GPL-3.0, AGPL-3.0 属于限制型许可证, 可以看出宽松型许可证占主导地位 (总计 76%), 限制型许可证次之 (21%), 而弱限制型许可证并不常见 (3%).

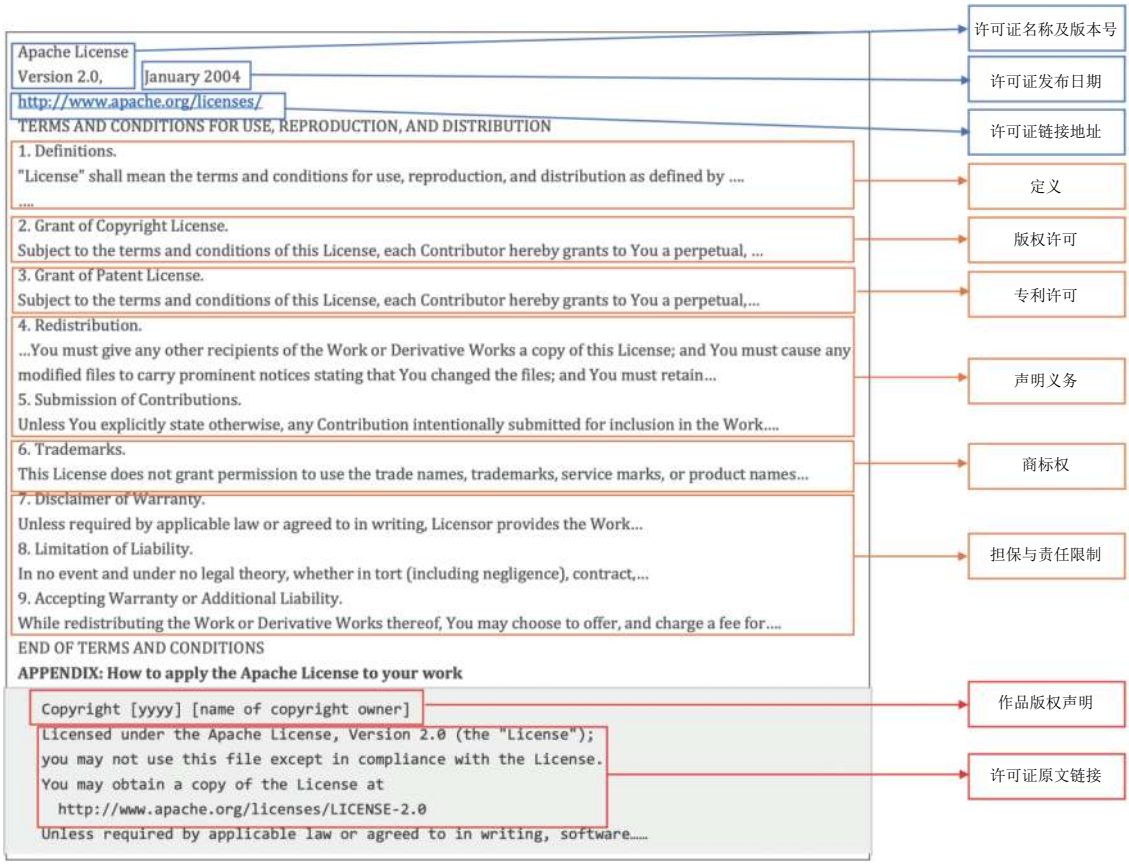


图 2　定性分析材料 (截取部分 Apache 2.0) 数据编码步骤示例

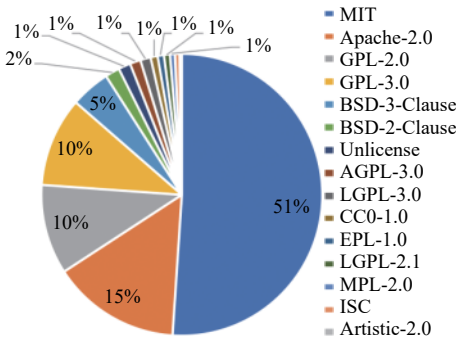


图 3　GitHub 上开源许可证使用现状

随后我们对选取的 10 种许可证的内容进行主题分析, 将许可证条款划分为 10 个维度: ① 许可证的基本信息, ② 序言, ③ 定义条款, ④ 授权条款, ⑤ 义务条款, ⑥ 违约与授权终止条款, ⑦ 担保与责任限制条款, ⑧ 准据法条款, ⑨ 许可证版本与兼容性, 以及⑩许可证使用说明等. 其中①许可证的基本信息、④授权条款中的版权许可、⑤义务条款、⑦担保与责任限制条款等为开源许可证中的常见条款, 其他条款通常根据许可证制定方的需求进行相应说明. 具体如下:

1) 基本信息. 主要包括许可证名称及版本号、发布日期、许可证版权声明及链接地址等信息.

2) 序言. 主要对许可证的适用场景或条件以及目的宗旨等进行说明, 如 GPL-3.0 的序言部分.

3) 定义. 为了便于开发者或用户理解许可证内容, 对许可证条款中的特定术语进行说明.

4) 授权. 开源许可证主要涉及的知识产权主要包括版权、专利权、以及商标权.

● 版权许可. 项目开发者通常免费授予用户行使相关权限, 包括使用、复制、修改、分发其开源项目或修改后的项目.

● 专利许可. 如果开源项目中包含专利, 开发者可以提供专利许可, 也可以不提供专利许可. 对于不提供专利许可的场景, 许可证将不会提及专利许可, 或者明确排除专利许可.

● 商标权. 开源许可证原则上不涉及商标权许可, 并且通常禁止借项目开发者的名义进行广告或宣传.

5) 义务. 用户在使用、复制、修改或分发软件或其衍生软件应当遵守的行为规范.

● 使用/复制/修改. 开源许可证通常不对使用目的、范围进行限制, 开发者可以基于任何目的 (学习、研究或商业) 对软件进行运行、备份、修改等操作, 但在一些特定场景中可能要求履行相关义务. 如 AGPL-3.0 中对通过网络使用软件向第三方提供服务时, 需要提供完整源代码.

● 分发. 当开发者分发软件或衍生软件时通常要求: ① 履行一定的声明义务, 例如修改声明 (如 Apache-2.0)、保留版权及免责等声明、提供许可证副本等; ② 要满足开源许可证对分发衍生软件的限制性, 根据限制性的强弱, 可以将开源许可证分为 3 类, 即宽松型 (permissive)、弱限制型 (weak-copyleft)、限制型 (copyleft).

6) 违约与授权终止. 对于用户违反许可证的行为, 可以对其终止授权, 也可以给予一定的补救机会.

7) 担保与责任限制. 项目开发者通常不对用户提供任何担保以及承担任何赔偿责任; 如果开发者个人对用户提供保证和担保须自行承担相应责任.

8) 准据法. 是指在许可证中指定援用的, 用来调整涉外民事法律关系双方当事人权利与义务的特定国家的法律.

9) 版本与兼容性. 对许可证版本进行说明, 以及对与该许可证兼容或者不兼容的其他许可证进行特别说明, 许可证兼容性是指项目中的许可证包含相互矛盾的必要条件, 而使得无法将其源代码合并成新的项目.

10) 使用说明. 告知用户将项目许可在该许可证下应该完成的步骤, 并可以包含一个声明模板, 对软件相关的作品描述、作品版权声明、许可证及其链接、作者联系方式等进行说明.

开源许可证之间的差异通常体现在③定义、④授权、⑤义务、⑥违约与授权终止以及⑧准据法中, 例如: MPL-2.0 和 CPAL-1.0 对“覆盖代码”定义的范围不同, 因此在下文条款中要求对“覆盖代码”使用原许可证分发其源码, MPL-2.0 可以通过不同文件来隔离传染性, CPAL-1.0 则需要不同模块单独分发来隔离传染性; 在授权中 Apache-2.0 明确提供授予专利权, 而 MIT 则未提及专利授权; 在义务条款中常见的差异主要来源于对分发衍生软件的限制性强弱以及不同的声明义务; 在授权终止中一些开源许可证中可能包含专利报复条款以及不同的违约补救条件, 如 Apache 中提到用户不得发起专利诉讼, 否则其授权将被终止; 此外, 还有一些开源许可证明确了准据法, 如 EUPL-1.2, 这也是开发者在使用开源许可证过程中需要注意的问题.

最后, 我们根据这 10 个维度, 提出了开源许可证框架 (图 4), 本文的作者通过对开源许可证进行交叉验证, 认为提取的开源许可证框架可以较好地解释目前开源许可证的结构组成, 能够帮助开发者理解许可证的条款.

4.3 结 论

我们发现, 目前在 GitHub 上广泛被使用的是 OSI 分类为“流行且广泛使用”的开源许可证, 其中宽松型开源许可证占主导地位, 限制型次之, 而弱限制型的开源许可证并不常见. 通过对开源许可证内容进行主题分析, 提出了

10 个维度的开源许可证框架, 其中, ①许可证的基本信息、④授权条款中的版权许可、⑤义务条款、⑦担保与责任限制条款等为开源许可证中的常见条款, 其他条款通常根据许可证制定方的需求进行相应说明. 开发者可以通过比较各个维度上开源许可证之间的差异, 来理解开源许可证的结构与含义. 此外, 开发者还可以利用开源许可证框架快速构建满足个人特定需求的新开源许可证, 例如, 木兰宽松许可证第 2 版 (MulanPSL-2.0) 正是在此开源许可证框架基础上为更好地保护开发者权益而构建的中国本土开源许可证.

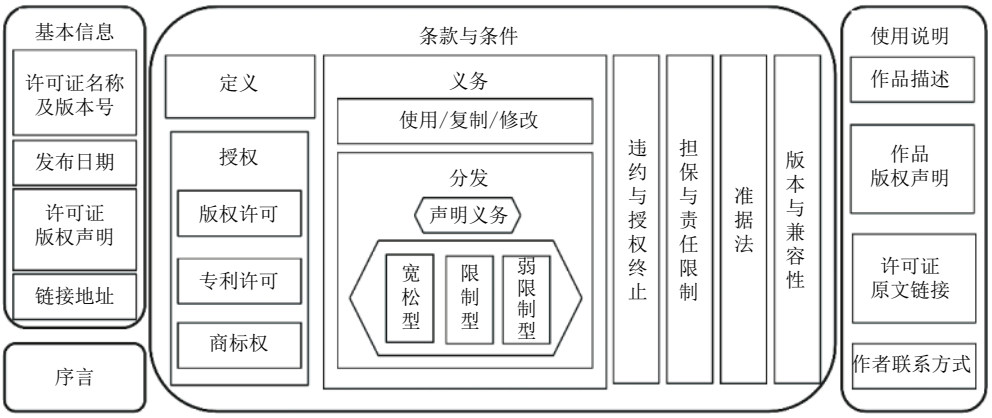


图 4 开源许可证框架

5 开源许可证选择的影响因素 (RQ3)

为了剖析影响开发者选择开源许可证的因素, 进一步为开发者结合自身需求选择合适的许可证提供决策支持和经验参考, 我们通过阅读有关文献并借鉴计划行为理论 (theory of planned behavior) 的 3 个维度设计调查问卷, 分析开源许可证选择的影响因素, 通过对项目特征因素拟合次序回归模型, 验证分析了项目特征与开源许可证选择的关系.

5.1 方法设计

(1) 问卷调研

在第 2.1 节中面向 200 开发者的调研问卷中的问题 4–问题 11 调研了开发者选择开源许可证的考虑因素. 其中, 问题 4 调研开发者选择开源许可证的影响因素, 共有 9 个选项 (多选, 详见表 2), 并增加一个 other (开放式回答) 选项, 收集开发者关于许可证选择因素的补充回答; 问题 5 (单选) 调研开发者具体支持何种开源理念; 问题 6 (单选) 调研开发者的首选开源许可证; 问题 7 (单选) 调研开发者如何看待开源商业; 问题 8 (多选) 调研开发者采用的商业模式; 问题 9 (多选) 调研开发者通常获得哪些开源利益; 问题 10 (开放式) 调研开发者是否为项目变更过许可证及变更原因; 问题 11 (开放式) 主要用于征询开发者补充关于开源许可证的更多建议和观点, 为进一步研究寻求突破口和补充我们问卷没有涉及到的问题. 在设计问卷问题过程中, 我们借鉴计划行为理论的 3 个维度, 结合有关文献, 初步分析了可能影响开发者选择开源许可证的因素. 计划行为理论是从信息加工的角度、以期望价值理论为出发点解释个体行为一般决策过程的理论, 大量研究证实它能显著提高研究对行为的解释力和预测力[43]. 在许可证选择过程中, 我们认为所有可能影响许可证选择的因素都是经由开发者的行为意向来间接影响最终许可证的选择. 这点经常被现有文献验证, 例如程序员的参与意愿强烈影响他在社区中的持续性[44]. 行为意向是指个人想要采取某一特定行为的行动倾向, 受到 3 项相关因素的影响: 一是个人本身的“行为态度”; 二是外在的“主观规范”; 三是“知觉行为控制”[45]. 为了保证问卷设计的合理性及计划行为理论维度的适用性, 在发送调查问卷前, 我们与 10 名有丰富开发经验的科研及企业人员进行了调研讨论, 详细地解释了研究目的和调研问卷设计的方法思路, 并明确要求他们填写问卷以检查问卷中可能包含的问题. 我们根据收集的建议对问卷中问题及选项的描述进行了澄清和改进. 实际问卷调研的 53 份回复不包含前述的实验性预调研结果.

表 2 开发者选择开源许可证的影响因素

(问题4)在选择开源许可证时, 您主要考虑的是什么?(多项选择题) (Q4)What are your main considerations when choosing an open source license? (multiple-choice)		
选项	数量	比例(%)
1. 取决于个人的意愿(According to your own will).	25	47.17
2. 考虑所采用的商业模式(Consider the business model used).	16	30.19
3. 所在组织的指导原则会影响许可证的选择(The guiding principles of your organization influence decision).	7	13.21
4. 考虑社区中开发者或用户的偏好(Preferences of developers or users from Community).	17	32.08
5. 考虑许可证的流行度, 是否被广泛使用(License popularity, whether such license is widely used).	13	24.53
6. 考虑许可证的法律复杂度(Legal complexity of the license).	12	22.64
7. 许可证之间的依赖, 依赖包的许可类型影响许可证的选择(License dependency, license choice may be limited to the license type of dependency packages).	9	16.98
8. 相似项目的许可证选择方案(Existing license choices for similar projects).	7	13.21
9. 考虑项目的特征, 如采用的编程语言、应用领域、项目规模等(Characteristics of your project (such as programming language, application domain, project size)).	9	16.98
10. 其他(Other).	5	9.43

问卷及 Q4 选项设计的理由如下:

① 行为态度方面, Ajzen 等人提出的期望-价值理论[46]认为态度包括个人实行某种行为的重要信念以及对价值的评价. 自由软件哲学思想的传播在自由软件和 GPL 的普及过程中发挥了重要作用[30], 开发者通常选择最符合自身意愿的开源许可证, 例如, 支持限制型许可证的人们认为共享代码是一个理想的最佳实践, 可以允许程序员创建更高质量的软件[47], 因此我们设计了选项 1. 近年来, 人们对开源软件作为一种替代经济模式的兴趣日益浓厚, 一些公司通常会寻找新的方式来产生收入和降低成本, 越来越多的公司将开源作为一种商业策略来实现这个目标[48], 受此影响, 我们设计了选项 2.

② 主观规范方面, 是指个体在决策时感知的社会压力或者外界因素的影响. 当开发者不是以个人名义参与开源时, 通常可能受到来自组织或企业的影响, 例如, 开发者在 Linux 社区中使用 BSD 许可证是可以接受的, 但是他们在 BSD 社区中贡献 GPL 代码是一个大大的禁忌[37], 由此我们设计了选项 3. 贡献者和用户对许可证类型的不同偏好可能影响开发者选择开源许可证, 例如, 具有限制性许可证的项目吸引的主要是寻求高度内在动机的贡献者, 而具有宽松许可证的项目吸引的不仅是寻求内在动机的贡献者, 还有希望获得商业化潜在机会的贡献者[24], 根据这一点我们设计了选项 4. 许可证的流行度和复杂度也可能影响开发者是否采用该许可证, 一个众所周知的和受信任的许可证对于开发者和用户来说都更容易被接受, 而过于复杂和鲜为人知的许可证可能容易造成混淆和歧义[16], 所以, 我们设计了选项 5 和选项 6. 此外, 项目之间的依赖使得许可证的选择可能会受到其他许可证的限制, 当开发者试图将他人编写的代码集成到自己的项目中时, 他们需要了解集成代码所携带的许可证, 尤其是组合不同许可证下的代码时, 兼容性问题会变得格外复杂[31], 有鉴于此我们设计了选项 7. 先前的研究发现, 项目采用的许可证类型往往受到与其关系密切的其他项目采用的许可证类型的影响[34], 由此我们设计了选项 8.

③ 知觉行为控制方面, 是指个人预期采取某一特定的行为时所感觉可以控制的程度, 常反映个人过去的经验、拥有的资源、能力以及预期的阻碍等. 从第 3.2 节中我们得知, 开发者在为项目选择许可证时会考虑不同的受众目标, 且有文献表明开发复杂软件的项目更可能选择有某种程度限制的许可证[8], 开发者可能依据不同的项目特征选择不同的开源许可证, 例如, Oracle 公司在 GitHub 上托管的不同开源项目中采用了多种开源许可证, 如微服务框架 Helidon 使用 Apache-2.0 开源, 而另一款移动交友软件 DinoDate 则使用 MIT 许可证开源, 有鉴于此我们设计了选项 9. 此外, 已有研究表明开源许可证的选择与项目的开发活动有关[12], 一定程度影响项目的发展. 开发者可以通过判断项目的发展趋势是否符合自己的意愿, 或者为了使许可证的选择满足自己变化的需求, 采用更换开源许可证以达到其目的, 因此我们设计了一道开放式问题 5 (开放式) 调研开发者是否为项目更换过许可证和更换

原因. 同时, 为进一步了解开发者的具体偏好, 我们针对上述选项 1 设计了问题 6 (单选, 表 3): 关于开发者具体支持何种开源理念的问题, 及问题 7: 开发者首选的开源许可证; 针对上述选项 2 (option2) 设计了问题 8 (单选, 表 4): 关于开发者如何看待开源商业, 问题 9 (多选, 表 5): 开发者采取了哪些商业模式, 以及问题 10 (多选, 表 6): 开发者从开源中通常获得哪些开源利益等问题.

表 3　开发者的开源理念

(问题5)你最认同哪一种观点?
(Q5) Which viewpoint do you agree with most?

选项	数量	比例(%)
1. 软件应该对所有用户自由的, 它可以被每个人共享和修改(Software should be free to all users, it could be shared and modified by everyone).	19	35.85
2. 用户有更多的权利以及更少的限制(Users might have more rights and fewer restrictions).	13	24.53
3. 介于两者之间(In-between).	21	39.62

表 4　开发者如何看待开源商业化

(问题7)您如何看待开源软件的商业化?换句话说, 就是通过使用开源软件来获得经济利益.
(Q7) What do you think about the commercialization of open source software? In other words, to gain economic benefit by using open source software.

选项	数量	比例(%)
1. 支持. 它可以促进开源的发展(Support. It could promote open source development).	37	69.81
2. 反对. 这是不道德的(Oppose. It would be immoral).	7	13.21
3. 介于两者之间(In-between).	9	16.98

表 5　开发者采用的商业模式

(问题8)你采用的开源业务模式是?
(Q8) Your open source business model? (multiple-choice)

选项	数量	比例(%)
1. 无(None).	14	30.43
2. 双重授权, 提供定制服务(Dual authorization, providing customization service).	14	30.43
3. 提供技术培训或售后服务(Provide technical training or after-sales service).	15	32.61
4. 生产基于开源软件的互补产品(Complementary products based on open source software).	20	43.48
5. 闭源销售(Closed-source commercialization, close source code, and sell software).	3	6.52
6. 获得声誉(Win the reputation).	12	26.09
7. 其他(Other).	1	2.17

(2) 定量分析

大量的工作探究了开源许可证类型与项目之间的关系[7,12,24], 也有研究利用相似用户或相似项目为开发者推荐开源许可证[18], 项目特征可能是影响开发者选择开源许可证的重要因素之一. 为了验证项目特征与许可证选择有关, 我们通过拟合次序回归模型分析了项目特征与许可证类型之间的关系, 通过数据分析及有关文献调研解释了可能原因, 为开发者提供参考借鉴. 具体步骤如下:

已有研究提出了几个重要的项目特征, 包括项目年龄、目标受众、编程语言、项目大小等[7], 我们对第 2.1 节

中选取的 4 704 个项目仓库, 提取出项目的创建日期、编程语言、应用程序描述、项目大小、许可类型、以及项目的开发数据等有关信息. 我们共设置了 4 类自变量, 分别为编程语言、应用领域、项目规模和项目年龄. 其中, ① 编程语言 (PL): 包括 C, C#, C++, Java, JS, Objective-C, PHP, Python, SQL, VB; ② 应用领域 (Domain): 通过人工分析项目的应用程序描述, 我们根据不同的受众, 将项目应用领域分为 6 类: 软件开发类 (Develop, 包括开发工具、库/框架等)、终端应用类 (App, 包括桌面应用、Web 应用、移动应用等)、流行技术类 (Popular, 包括人工智能、云计算、数据科学、区块链等)、底层相关类 (Underlying, 包括操作系统、数据库、中间件、硬件相关等)、教程 (Tutorial) 及游戏 (Game). ③ 项目规模 (Size): 我们利用 GitHub 上项目的存储大小信息按照分布的百分位数分为 3 个层次: 其中前 1/3 为小项目 (≤1 MB), 中间 1/3 为中项目 (1–20 MB), 后 1/3 为大项目 (>20 MB); ④ 项目年龄 (Age): 以月为计量单位, 按照创建时间的先后进行统计. 因变量为许可证类型 (LicenseType=1, 2, 3), 分为宽松型, 弱限制型以及限制型. 自变量中的编程语言和应用领域为无序分类变量, 使用两组虚拟变量表示, 其中编程语言中的 VB 和应用领域中的游戏 (Game) 为参照类, 我们将上述变量带入次序回归模型进行拟合. 次序回归模型定义如下式:

$$\text{LicenseType} \sim \text{PL} + \text{Domain} + \text{Size} + \text{Age}.$$

表 6　开源的非经济利益

(问题9)开源给你带来什么好处?(多项选择题) (Q9) What benefits does open source bring to you? (multiple-choice)		
选项	数量	比例(%)
1. 个人声望(Personal reputation).	43	81.13
2. 产品可见性(Product visibility).	31	58.49
3. 建立行业标准(Establishment of industry standards).	17	32.08
4. 挑战的乐趣(Challenge fun).	38	71.70
5. 职业发展(Career development).	28	52.83
6. 扩大用户基础(Broad user base).	22	41.51
7. 社区中开发者支持(Support of developers from the community).	31	58.49
8. 其他(Other).	2	3.77

5.2 结果分析

根据计划行为理论的 3 个维度结合有关文献调研, 我们提出影响开源许可证选择有 9 个因素, 其中, ① 开发者的开源理念、② 开发者对利益的评估等涉及行为态度方面; ③ 开发者所在组织观念的影响、④ 社区偏好的影响、⑤ 许可证流行度和复杂度、⑥ 许可证兼容性、⑦ 其他项目的影响等涉及主观规范方面; ⑧ 对项目特征的评估、⑨ 许可证选择结果的影响等涉及知觉行为控制方面. 问卷调研结果验证了各个影响因素的相关性. 我们进一步通过拟合次序回归模型验证了项目特征与开源许可证选择的关系, 具体分析如下.

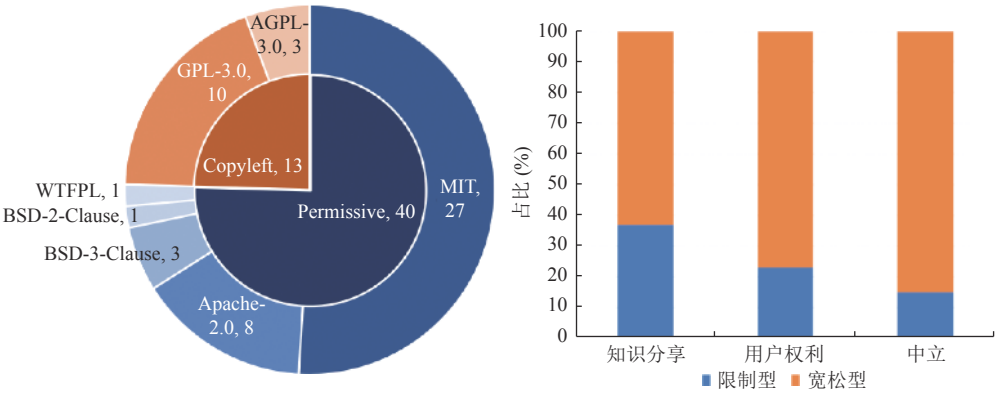
① 开发者的开源理念

开源理念反映了开发者如何看待开源软件的使用问题, 当开发者认为某一个开源许可证所蕴含的开源哲理与其开源理念相符时, 更有可能表现出强烈的选择意愿. 问题 4 选项 1 反映了开发者的开源理念 (表 2), 调研中发现 47.17% 的开发者认为个人意愿是影响他们选择开源许可证的一个因素. 此外开发者还在补充回答中提到“自由/开源哲学理念 (FLOSS philosophy)”“为了防止其他人将我们的项目用于商业产品, 最大化耗费开源资源 (To prevent others taking our project and using it in a commercial product, maximize consumption)”. 也进一步说明了开源理念在许可证选择过程中的重要影响.

关于如何开放源码, 不同的开发者有不同的理念, 在问题 5 的调研 (表 3) 中, 我们得到在开发者支持何种开源理念的调研结果中, 35.85% 的开发者支持知识共享观念, 一定程度上说明他们中许多人不希望源代码被第三方私

有化, 而是期望获得最大的贡献和反馈[2], 可能更愿意使用限制型许可证; 24.53% 的开发者则支持更多最终用户权利的开发人员更喜欢限制较少的开源许可证[8]; 而 39.62% 的开发者没有明显的倾向性, 他们更有可能受到其他一些因素的影响.

在问题 6 (Q6) 的关于开发者首选开源许可证的调研结果 (图 5(a)) 中, 显示 75.47% 的开发者选择了宽松型开源许可证, 25.53% 的开发者选择了限制型开源许可证, 而没有开发者选择弱限制型开源许可证, 总体上与 GitHub 数据集观察到的许可证使用情况的分布相吻合. 我们通过交叉图 (图 5(b)) 分析开发者支持的开源理念与其首选开源许可证, 发现支持知识共享的开发者相比于支持更多用户权利的开发者更容易选择限制型的开源许可证. 然而, 开发者的开源理念并不完全决定其选择的开源许可证类型, 一部分支持知识共享观念的开发者选择了宽松型开源许可证, 而另一部分支持更多用户权利的开发者也选择了限制型的开源许可证, 说明开发者选择开源许可证时通常受到除开源理念以外其他方面因素的影响.



(a) 开发者 选开源许可证 (b) 开发者 选开源许可证与其开源理念交叉图

图 5 开发者首选的开源许可证调研结果

② 开发者对利益的评估

开源并不一定出于利他主义或意识形态, 它也可以出于健康的自身利益[34]. 开发者参与开源获得的利益可以分为经济利益以及非经济利益.

一是经济利益方面, 虽然开源软件第一眼看上去似乎是反利润的, 但越来越多的企业或组织已经将开源思想纳入到他们的商业战略[49]. (表 2) 问题 4 选项 2 的结果显示 30.19% 的开发者认为其采用的商业模式是选择开源许可证的考虑因素. 在问题 7 关于开发者如何看待开源商业的调研结果中 (表 4) 发现, 69.81% 开发者支持开源商业, 认为可以促进开源的发展, 也有 13.21% 开发者反对通过开源获得经济利益, 认为这是不道德的, 而 16.98% 的开发者保持中立态度.

针对支持开源商业和保持中立态度的 46 名开发者, 我们继续调研他们是否采用以及采用了何种商业模式 (表 5), 调研发现, 其中 69.57% 的开发者在实际开发中采用了商业模式, 包括: 双重授权、支持服务、互补产品、闭源销售和赢在声誉等. 应用最广泛的是生产互补产品 (43.48%), 是指基于开源的计算或服务项目提供商业的插件或配套硬件等, 如, 销售 CD 版的 Linux, 销售 Eclipse 的商业插件[50]. 其次是支持服务 (32.61%), 是指针对开源软件项目为客户提供支持、维护、开发、咨询或培训服务, 或提供与开源相关的审计和法律服务等, 例如 mLab 为用户提供数据库项目的托管服务, RedHat 为付费用户提供“知识产权保障计划”[50]. 再次是双重授权 (30.43%), 通常包含两种方式: 一是在开源许可下提供有限的或精简版的软件产品, 而在专有许可证下提供增强的或升级版的软件产品[50], 开源免费版主要用于扩大产品可见性和知名度, 例如开发工具 Pycharm 同时提供社区免费版以及增强功能的企业版; 二是针对同一个项目同时提供开源许可和商业许可, 这里采用的开源许可证通常具有 copyleft 特性, 主要是为了扩大用户群及降低竞争对手生产盗版私有产品的机会, 而为那些希望不受 copyleft 影响的用户

提供商业许可, 例如 MySQL 和 Sleepcat 的商业策略. 而后是赢在声誉 (26.09%), 为了推广技术或在行业中建立新的标准, 将其项目开源, 不仅有利于后续产品的推广, 还可以从商标、广告中获取经济利益. 最后是闭源销售 (6.52%), 是指将开源软件与私有软件相结合, 并作为私有软件销售, 如 AWS 提供了 Apache Hadoop 软件的商业版本, 然而仅靠销售软件获利并不容易, 开源社区已经提供了该产品的免费版本, 只有能够为产品增加相当大的价值时, 才能产生可观的利润[48].

二是非经济利益方面, 开发者参与开源还可能有经济利益以外的目的, 开发者可以通过选择不同的许可证类型以期获得不同的非经济利益, 例如限制型许可证可以提供更高的贡献可见性, 开发者更容易获得期望的认可、声誉或职业机会[29], 而宽松型许可证对用户没有限制, 容易获得更大的用户群[37]. 问题 9 调研发现 (表 6), 开发者通过开源获得的非经济利益主要包括提升个人名誉 (81.13%)、获得挑战乐趣 (71.7%)、社区开发者的技术支持 (58.49%)、职业发展 (52.83%)、提升产品可见度 (58.49%)、建立行业标准 (32.08%)、广泛的用户基础 (41.51%)、还有开发者提到“我还用开源库发表了论文 (I also published the libraries as papers)”“企业间合作的投资安全 (Safety of investment for collaboration between companies)”.

③ 开发者所在组织观念的影响

当开发者代表组织或在特定社区进行开源时, 通常也需要考虑所选择的许可证是否符合其所在组织的观念, 而避免不必要的纠纷, 例如上文中提到的在 BSD 社区中应避免选用 GPL 类开源许可证. 表 2 选项 3 的结果显示 13.21% 的开发者认为其所在组织的观念可以影响其选择开源许可证.

④ 社区偏好的影响

不同的开源许可证类型不同程度地吸引开源社区中的贡献者和用户, 例如: 宽松型许可证允许与私有软件合并, 因此对商业用户更有吸引力, 而一些社会性质项目的开发者可以通过选择限制型许可证来吸引更多的开发人员[8], 因为限制型许可证往往产生更强的社会认同感[15]; Sen 等人还发现, 受到工作挑战激励的人更喜欢有适度限制的许可证, 而那些重视诸如地位或机会之类外在动机的人更喜欢宽松的许可证[8]. (表 2) 问题 4 选项 4 结果表明 32.08% 的开发者认为贡献者和用户对不同类型许可证的偏好影响其选择开源许可证,

⑤ 许可证流行度和复杂度

(表 2) 问题 4 选项 5 结果表明 24.53% 的开发者慎重考虑了开源许可证是否被广泛使用, 开发者通常被建议使用现有的经过实践检验的许可证, 而不是起草一个新许可证. 而选项 6 结果表明 22.64% 开发者认为许可证法律复杂度是影响其选择开源许可证的重要因素, 例如开源许可证中准据法的条款使得开发者不得不考虑其知识产权被合理使用的范围.

⑥ 许可证兼容性

(表 2) 问题 4 选项 7 结果表明 16.98% 的开发者认为项目之间的依赖导致的许可证兼容性问题是其选择开源许可证时考虑的因素, 这种情况常常发生在使用了限制型开源许可证的项目. 开发者在补充回答中也明确指出“许可证的权限, 如项目的副本应以原始项目为核心, 采用相同的许可证 (Restrictions and permission of the license (copy of the project should core the original project, stays under the same license))”.

⑦ 其他项目的影响

(表 2) 问题 4 选项 8 结果表明 13.21% 的开发者为项目选择开源许可证时可能考虑与其类似项目或者是其他大型知名开源项目所使用的开源许可证. 例如, 开发者补充到“我只是模仿大型库的许可方式, 比如 scikit learn (I just mimic what big libraries are doing, e.g.scikit learn)”等.

⑧ 对项目特征的评估

(表 2) 问题 4 选项 9 的调研结果发现仅 16.98% 的开发者会根据项目的特征考虑不同的开源许可证, 一定程度说明项目特征因素在开发者实际选择开源许可证并不重要.

鉴于项目特征对开发者的重要性和可评估性, 我们分析了项目特征与开源许可证类型的具体关系, 结合文献调研解释其原因, 为开发者参考项目特征选择开源许可证提供参考. 我们通过对第 3.1 节中选取的 4 704 个项目进行定量分析. 我们从项目中提取编程语言、应用程序描述、大小 (单位为 KB)、创建时间和许可证信息, 并根据

5.1(2) 中对变量设定原则得到编程语言 (PL)、应用领域 (Domain)、项目大小 (Size)、项目年龄 (Age) 和许可证类型 (LicenseType) 等信息 (项目在各变量维度上的分布如图 6、图 7、图 8 所示), 并进行相关关系分析 (表 7) 及次序回归拟合 (表 8). 回归模型通过了平行线检验 (表 9), 说明模型是有效的, 项目特征与许可证类型有关, 伪 R^2 表示项目的 4 个特征对许可证类型的解释程度, R^2 (Cox and Snell) 值为 0.169, 说明开源许可证的选择还受到项目的其他特征或者项目特征以外的其他因素的影响, 与上述调研结果是相符的, 也一定程度地说明了仅仅通过相似项目为开发者推荐许可证是不全面的.

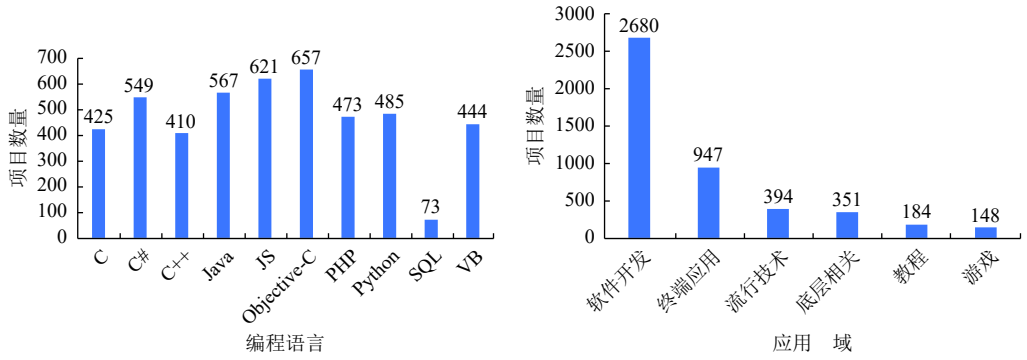


图 6 项目在编程语言 (左图) 和应用领域 (右图) 的分布

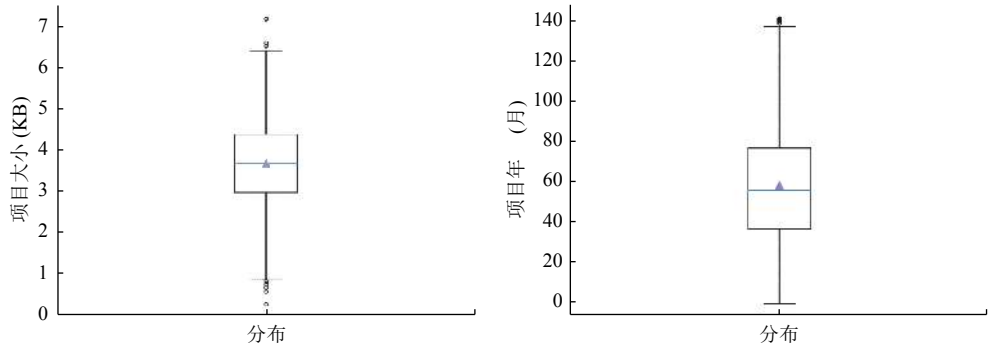


图 7 项目在大小 (左图) 和项目年龄 (右图) 上的分布

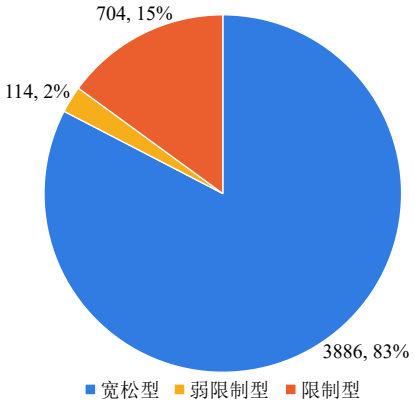


图 8 项目在开源许可证类型上的分布

从表 7 的相关系数和表 8 的回归模型数据中可以发现, 项目年龄与许可证类型的关系并不显著, 可以认为在近一年里许可证类型的使用分布情况变化不大. 项目规模与许可证类型的关系是正向显著的, 表明项目规模

越大, 越容易选择限制较强的许可证, 正如 FSF 的常见问答中也建议对于代码量较小的项目可以使用限制较少的许可证, 而且项目规模越大, 开发者可能投入了更多的努力, 限制型许可证可以更好地保护开发者的努力不被第三方获取.

从编程语言和应用领域这两个分类变量的置信区间估计值及回归系数 (表 8) 可以观察到, 软件开发类、流行技术类、教程等领域更容易选择宽松型许可证, 而在终端应用类、底层相关类、游戏等领域限制型许可证比在前述其他领域中的限制型许可证更常见. 从编程语言来看, 编程语言 C、C++、SQL、VB 相比于其他语言 C#、Java、JS、Objective-C、PHP、Python 更容易选择限制型许可证. 正如我们所了解的, C、C++通常用于服务端的服务程序开发、硬件和系统开发等, 而 SQL 用于数据库领域, VB 主要用于游戏开发以及对软件进行二次开发, 相比于 Java、Python、C#等倾向于应用开发的语言, 前者更偏向于面向底层开发或者其面向特定的受众. 交叉分析统计图 (图 9) 可以直观对比不同类型的开源许可证分别在编程语言和应用领域上的差异, 可能的原因是, 一方面由于终端用户很少需要源代码, 而对于大众市场的软件开发人员通常把核心产品的源代码作为公司盈利的宝石, 使用限制性较强的许可证可以避免竞争对手从访问源码中获得搭便车的好处[47], 另一方面对于底层相关领域的技术实现相对复杂, 开发者通常不希望源码被第三方私有化, 且采用限制型许可证可以避免因克隆产生过多分支版本而影响原版本的发展和权威.

表 7 相关系数

项目特征	M	SD	1	2	3	4	5
1. Age	58.36	27.830	1	−0.067**	−0.209**	0.032*	−0.007
2. PL	5.18	2.648	−	1	0.032*	−0.246*	−0.004
3. Domain	1.91	1.341	−	−	1	0.092**	0.229**
4. Size	2.02	0.730	−	−	−	1	0.060**
5. LicenseType	1.32	0.720	−	−	−	−	1

注: 列标题中1,2,3,4,5分别表示项目年龄、编程语言、应用领域、项目规模和许可证类型. *p<0.05.**p<0.01.***p<0.001

表 8 参数估计值

项目体征		LicenseType		95%CI	
		coefficient	wald	upper	lower
Age		0.006	13.425	0.003	0.009
Size		0.320***	27.498	0.201	0.440
PL	C	0.040	0.068	−0.260	0.339
	C#	−1.365***	62.205	−1.704	−1.026
	C++	−0.308	3.661	−0.624	0.008
	Java	−1.921***	88.197	−2.322	−1.520
	JS	−2.635***	115.166	−3.117	−2.154
	Objective−C	−2.180***	108.830	−2.590	−1.771
	PHP	−0.966***	28.856	−1.318	−0.613
	Python	−0.867***	26.086	−1.200	−0.534
	SQL	−0.344	1.408	−0.913	0.224
	VB	0a	−	−	−
Domain	Develop	−1.159***	34.861	−1.543	−0.774
	App	−0.472*	6.051	0.096	0.849
	Popular	−0.998***	17.500	−1.466	−0.531
	Underlying	−0.295	1.800	−0.726	0.136
	Tutorial	−0.954**	10.150	−1.541	−0.367
	Game	0a	−	−	−

注: *p<0.05.**p<0.01.***p<0.001

表 9　模型平行线检验

指标	参数	值
自由度	df	16***
样本数	N	4 704
R^2	Cox and Snell	0.169
	Nagelkerke	0.258
	McFadden	0.174

注: *p<0.05.**p<0.01.***p<0.001

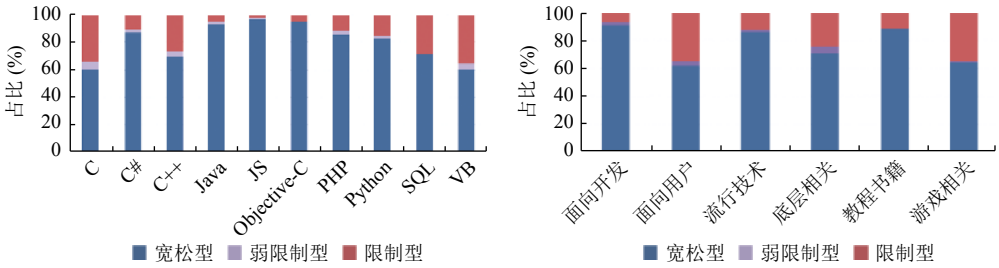


图 9　开源许可证不同类型在编程语言 (左图) 和应用领域 (右图) 的使用分布

⑨ 许可证选择结果的影响

由于社区的贡献者和用户对开源许可证类型存在不同偏好, 一定程度影响着他们是否参与或接受该项目, 且不同开源许可证的类型可以带来不同的开源利益, 从而对项目的发展带来一定的影响. 一方面, 开发者可以通过分析其他项目许可证选择结果的影响, 为自己选择开源许可证提供决策支持; 另一方面, 开发者可以根据个人开源项目发展趋势或经验进行判断, 调整对开源许可证的选择. 我们在问题 10 调研了他们是否为项目变更开源许可证及变更原因, 53 名回复者中有 7 名开发者 (13.20%) 回复了该问题, 表明他们为项目更换过许可证, 主要原因包括支持项目的商业化、使项目更容易被接受以扩大用户群、通过更强的限制以达到保护项目不被商业化的目的, 以及尽量避免项目因许可证涉及法律纠纷等 (“I have changed from BSD license to Apache license as the former one is more suitable for EU laws”“I have modified some projects, mainly “Libraries”, from GPL-v3 to LGPL-3.0 to ease project acceptance.”“Switched from GPL-2.0 after learning about its complexities and issues”“We had things in GPL that we moved to MIT as that is useable in commercial projects”“I changed from GPL to AGPL to protect my application’s API’s as well.”“To broaden the consumer audience”“Switched from closed source to open 5 years ago.”). 而前文中提到的关于 Redis 变更自研模块的许可证, Redis Labs 的联合创始人兼首席技术官 YiftachShoolman 表示变更的目的是保护开发者的利益和开源软件的持续发展. “多年来, 云提供商通过销售基于开源项目的云服务, 可从中获利数亿美元, 可这些项目实际上并不是他们自己开发的, 如 Docker, Elasticsearch, Hadoop, Redis 和 Spark. 这阻碍了社区投资开发开源代码, 因为任何潜在的好处都归云提供商而不是代码开发人员或他们的赞助商[4]”.

此外, 问题 11 收到 9 个回复. 其中, 部分回复的内容与开源许可证的选择无关, 例如, “希望你们可以提供一篇选择开源许可证的参考, 方便我们这些开源的朋友学习!”“You can read all about it here: https://www.computerweekly.com/blog/Open-Source-Insider/How-to-create-a-successful-open-source-business-model”“* Public money must always turn into public code * It’s important for OS adopters to understand the alternative cost of their choice (how much would have cost them to develop that piece of OS code) * Market competency is the main reason of OS license violation”.

同时, 部分回答与问卷中前述问题的回复观点重复, 例如, “MIT hydra”“I prefer MIT for its simplicity and usage freedom”“GPL-2.0 is still the most popular, and has the most benefit to the developer. Liberal licenses like MIT and Apache mostly benefit hardware and systems vendors”“I believe in open source software. By choosing (A) GPL over

MIT or Apache, I force other users to share their work. Anybody could make money from my software, even if they have to share their improvements (make it ready for many consumers). But they never do this. They don't care about open source. Just want to profit off my work”“We chose GPL v3 specifically because we invested a lot of work in our software, and we did not want others to simply take our work, rebadge it, and commercialize it. We also make the software available under a custom proprietary license, for those who want it”.

因此，我们过滤了与开源许可证的选择无关的回答, 而对于与问卷前述问题观点的重复的内容, 我们将其归到了 Q6 和 Q4 的结果之中, 这几名受访者分别在 Q6 和 Q4 补充回答中的回复与其在 Q11 中的回复观点相同.

5.3 结　论

我们发现, 开发者的开源理念、对利益因素的评估、开发者所在组织的观念、开源社区对许可证的偏好、许可证流行度和复杂度、许可证兼容性、其他项目的影响、开发者对项目特征的评估, 以及许可证选择结果的影响, 都可能在某种情况下影响开发者为项目选择许可证 (图 10), 且开发者选择开源许可证并非受这些因素中单个因素的影响, 而是受到多方面因素的影响. 首先, 行为态度方面是影响开发者选择开源许可证过程中最常见的影响因素, 其中支持 copyleft 观念的开发者相比支持更多用户权利的开发者更可能选择限制型开源许可证, 同时开源不仅可以为开发者带来声誉、认可等非经济利益, 还可以带来经济利益, 越来越多的企业和个人已经将商业模式应用到开源中, 包括生产互补产品、支持服务、双重授权、赢在声誉、闭源销售等, 开发者可以根据其对不同利益类型的偏好及所采用的不同商业模式考虑不同的开源许可证; 其次, 尽管开源许可证的选择是属于个人行为, 但开发者所处社会环境的影响也是其选择开源许可证重点考虑的方面, 例如所在组织的观念、社区偏好及开源许可证因素的影响; 最后, 大量研究关注于开源许可证与项目之间的关系, 然而我们通过调研和定量分析发现, 开发者实际选择开源许可证的过程中, 项目特征一定程度影响选择结果, 但其并不是必然考虑的因素, 开源许可证选择工具需要从开发者实际需求出发, 通过综合判断分析, 才能为开发者推荐适用的开源许可证.

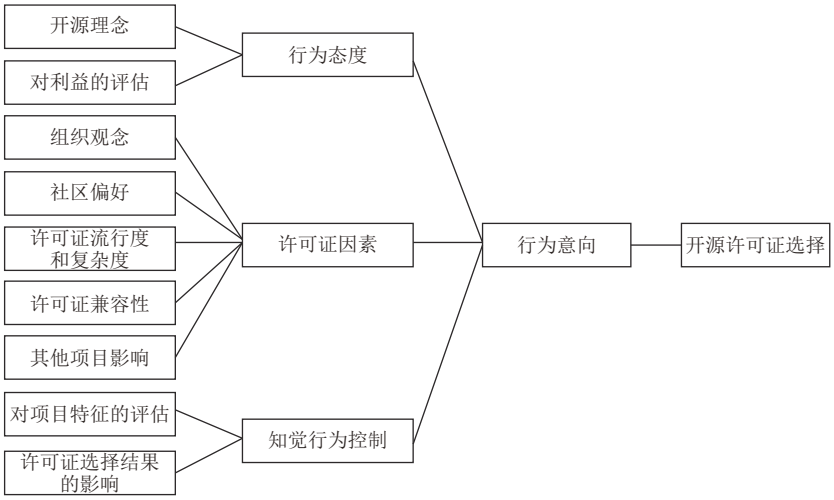


图 10　开源许可证选择的影响因素

6 讨　论

本文针对开发者难以根据自身需求选择合适的许可证的现状, 研究了以下问题: ① 开发者为项目选择开源许可证时通常会面临哪些困难? ② 开源许可证的组成要素有哪些? ③ 哪些因素影响开发者选择开源许可证? 我们通过问卷调研 200 名 GitHub 开源开发者, 获得了开发者选择开源许可证通常面临的两类困难和 9 大影响因素, 并通过分析 GitHub 开源项目中使用最广泛的 10 种开源许可证, 建立了一个开源许可证框架, 我们还针对开发者普遍

关心的项目特征这一影响因素, 通过对项目特征因素拟合次序回归模型, 验证了项目特征与开源许可证选择的关系.

我们建立的开源许可证框架可以帮助开发者理解开源许可证的构成和差异. 刚刚接触开源的开源开发者通常缺乏法律知识背景, 开源小企业可能没有专业的法务团队支持, 开源许可证之间的差异以及条款中包含的法律词汇常常使他们感到困惑, 开源许可证框架可以帮助开发者基于 10 个条款维度分析开源许可证, 对理解不同开源许可证之间的差异是十分有效的, 例如在专利授权条款中有无对违反专利授权的限制可以体现出该开源许可证对开发者利益保护程度、根据在分发义务条款中对衍生作品分发的限制可以将开源许可证划分为宽松型、弱限制型和限制型等. 开源开发者或开源企业还可以利用开源许可证框架快速构建满足自身需求的开源许可证. 尽管开源社区已有大量不同类型的开源许可证, 这些开源许可证仍然可能无法满足开源开发商的所有需求, 他们试图构建自己的开源许可证, 例如, MongoDB 改用一种新的服务器端公共许可证 (SSPL) 力求堵住一些云提供商利用其开源代码生产数据库的托管商业版本而无需开源的缺口 (https://www.sohu.com/a/260120443_465914). 因此, 我们提出的开源许可证框架还可以帮助开发者、社区或开源企业快速构建满足自身需求的开源许可证, 在此框架基础上构建的木兰宽松许可证, 第 2 版 (MulanPSL-2.0) 通过 OSI 认证也进一步证明了该框架的有效性和通用性.

我们揭示的影响开源许可证选择的 9 大因素可以帮助开发者或开源企业从不同角度全面分析自身的需求来指导开源许可证的选择. 不同类型的开源许可证可以带来不同的开源利益, 且一定程度影响贡献者和用户是否参与或接受该项目, 从而对项目的发展带来一定的影响. 开源企业参与开源的目的与其需求直接相关, 他们通常依据自身不同需求针对不同的开源项目在不同的应用场景下选择不同的开源许可证, 例如, 在一篇名为《分析了 35 家企业 75 个开源项目》的译文中重提到, 唯一使用相同许可模式的公司只有 Palantir (硅谷一家数据挖掘公司)[51]. 因此, 企业对自身业务需求的准确把握需要综合考虑各方面因素, 这对于开源许可证的选择至关重要. 本文结合问卷和文献调研得到的影响开发者选择开源许可证的 9 个因素反映了开发者选择开源许可证不同方面的考虑, 我们建议开发者可以从 9 个影响因素的维度全面准确分析自身需求, 在充分理解开源许可证条款的基础上做出综合判断, 从而选择符合自身需求的开源许可证.

此外, 我们的研究结果还可以帮助业界清晰了解开发者选择开源许可证面临的实际困难和影响因素, 采用更好的策略来解决开发者遇到的困难, 例如改进现有的开源许可证选择工具来帮助开发者结合自身需求选择合适的许可证提供决策支持. 目前业界和学术界实现的基于开源许可证条款差异 (OSSWATCH Licence Differentiator)、基于简单应用场景 (ChooseALicense)、或基于相似项目推荐[18]的开源许可证选择工具, 没有全面考虑到开发者的实际困难和需求, 难以帮助开发者选择适用的开源许可证. 我们结合定量分析和调研也发现尽管项目特征和开源许可证类型存在一定的相关性, 但其并不是开发者选择开源许可证时必然考虑的因素, 开源许可证选择工具需要从开发者实际需求出发, 通过综合判断分析, 才能为开发者推荐适用的开源许可证. 本文得到的 9 个影响因素维度为实现基于用户需求的开源许可证选择工具提供了思路, 通过识别开发者的需求, 借鉴相同或相似需求的成功或著名开源项目所选择的开源许可证, 为开发者推荐适用的开源许可证. 我们的未来研究方向是如何获取和利用开发者的不同需求, 结合不同开源项目的特征和应用场景, 为开发者推荐合适的开源许可证.

7 局限性

论文的局限性主要体现在数据的有效性、方法的构造性和结论的普适性上面.

第 1 个局限性是数据的可访问性和一致性. 首先, 本文通过 GitHub search API 爬取的项目数据只包含有限项目仓库, 我们获取了编程语言流行度排名前十的总共 9672 个项目仓库, 这些项目仓库包含了近一年的数据, 一些仓库没有公开或者无法通过 GitHub search API 获得, 在数据清洗过程中, 我们去掉了许可证为空或 Other 的项目, 因其无法提取项目的许可证信息; 其次, 数据本身可能无法反映实际的情况. 在第 5.2 节中我们定量分析了项目特征与许可证类型之间的关系, 其中项目的特征仅考虑了编程语言、应用领域、项目规模和项目年龄, 可能存在其他的特征影响着许可证的选择, 例如项目的技术复杂性等. 尽管这可能导致结果分析存在一定的局限性, 但编程语言的流行度与软件开发领域发展和市场热度息息相关, 通常是开发者关注的热点领域, 而对最近一年数据的开源

许可证使用情况分析也正好反映开发者选择开源许可证的现状. 因此, 我们认为所获取的数据能够适用于本文研究工作的目的.

第 2 个局限性是方法的构造性. 首先, 本文主要采用主题分析方法和计划行为理论等定性分析方法对调研结果进行分析, 然而采用定性研究结果指导实践具有一定的局限性; 其次, 在分析开源许可证选择影响因素时, 我们认为行为意向能较好地解释开发者选择开源许可证的行为, 因此借鉴了计划行为理论的 3 个维度设计调研问卷, 然而从计划行为理论的研究进展可以发现, 计划行为理论的主要变量的概念定义一直是研究者们争论的焦点[42]. 这可能对分析结果的准确性造成一定的影响. 我们通过预调研的方式确保问卷设计的合理性及计划行为理论维度的适用性, 并对定性分析的结果进行交叉验证从而减少这类影响.

第 3 个局限性是结论的普适性. 在调研中随机选取的 200 名开发者作为调研对象不能代表所有开发者的意见, 由于邮件调研的特殊原因, 一部分开发者的邮箱地址为公司客服邮箱或者已注销或不再使用, 难以得到有效的回复, 且开发者容易受到当时环境、心理等因素的影响, 可能存在其他未调研到的实际困难和影响因素, 使研究结果的应用范围和应用程度受到一定的局限. 由于国际邮件调查的回复率普遍较低, 在软件工程领域，邮件回复率通常在 6%–36%[42], 而我们的回复率在 25%, 是一个相对较高的回复率. 同时, 我们通过阅读大量文献、与相关企业开发人员访谈设计问卷、预调研等多种方式确保问卷的合理性和选项设计涵盖的普遍性, 通过实际问卷调研来获取更广泛的反馈并细致分析结果, 最终得到开发者选择开源许可证时面临的困难和影响因素. 我们认为所得研究结果可以反映当前开发者普遍面临的困难及其选择开源许可证过程中主要的考虑因素, 具有普适性.

8 总 结

本文首先通过问卷的方式调研了开发者在为项目选择许可证时通常会遇到的两类困难, 即因许可证的相似性和法律复杂性造成开发者难以理解开源许可证之间的差异, 以及开发者对如何全面考虑各方面因素进行最佳决策感到困惑, 从而有助于清晰理解开发者选择开源许可证面临的实际困难. 其次, 我们通过对比分析最广泛使用的十种开源许可证的条款, 将许可证内容划分为 10 个维度, 并建立起开源许可证框架, 可以帮助开发者清晰认识和理解开源许可证的内容构成, 便于开发者从各个维度出发对比和分析开源许可证之间的差异, 减轻开发者解决上文提到的第一类困难的压力; 同时, 开发者还可以利用开源许可证框架快速构建符合自己特定需求的新开源许可证. 最后, 我们通过对开发者选择开源许可证考虑因素的调研和分析, 得出影响开源许可证选择存在多方面因素, 包括开发者的开源理念、对利益的评估、组织观念和社区偏好的影响、许可证流行度或兼容性问题、其他项目的影响、个人对项目的评估以及许可证选择结果对项目的影响, 为开发者结合自身需求选择合适的许可证提供决策支持和经验参考, 可以帮助开发者解决上文提到的第二类困难, 为实现基于用户需求的许可证选择工具提供借鉴.

References:

[1] Zhang YX, Zhou MH, Mockus A, Jin Z. Companies' participation in OSS Development-An empirical study of OpenStack. IEEE Trans. on Software Engineering, 2019. [doi: 10.1109/TSE.2019.2946156]

[2] Kaminski H, Perry M. Open source software licensing patterns. Computer Science Publications, 2007, 10. https://ir.lib.uwo.ca/csdpub/10

[3] Lynch J. Linus Torvalds credits GPL with preventing Linux fragmentation. Info World, 2016. https://www.infoworld.com/article/3112778/linus-torvalds-credits-gpl-with-preventing-linux-fragmentation.html

[4] Darwin. Redis module open source license change, many projects no longer open source questioned. 2018. https://www.oschina.net/news/99271/redis-database-license-change (in Chinese).

[5] Cheng DJ. Open source fog of Android and opportunities for Chinese manufacturers. Communications World, 2013, (7): 11 (in Chinese with English abstract). [doi: 10.13571/j.cnki.cww.2013.07.003]

[6] Lin YH, Ko TM, Chuang TR, Lin KJ. Open source licenses and the creative commons framework: License selection and comparison. Journal of Information Science and Engineering, 2006, 22: 1–17.

[7] Colazo J, Fang YL. Impact of license choice on open source software development activity. Journal of the American Society for Information Science and Technology, 2009, 60(5): 997–1011. [doi: 10.1002/asi.21039]

[8] Sen R, Subramaniam C, Nelson ML. Determinants of the choice of open source software license. Journal of Management Information

Systems, 2008, 25(3): 207–240. [doi: 10.2753/MIS0742-1222250306]

[9] Vendome C, Linares-Vásquez M, Bavota G, Di Penta M, German DM, Poshyvanyk D. When and why developers adopt and change software licenses. In: Proc. of 2015 IEEE Int'l Conf. on Software Maintenance and Evolution. Bremen: IEEE, 2015. 31–40. [doi: 10.1109/ICSM.2015.7332449]

[10] Zhou MH. Onboarding and retaining of contributors in FLOSS Ecosystem. In: Fitzgerald B, Mockus A, Zhou MH, eds. Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability. Singapore: Springer, 2019. 107–117. [doi: 10.1007/978-981-13-7099-1_7]

[11] Lerner J, Tirole J. The scope of open source licensing. The Journal of Law, Economics, and Organization, 2005, 21(1): 20–56. [doi: 10.1093/jleo/ewi002]

[12] Stewart KJ, Ammeter AP, Maruping LM. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. Information Systems Research, 2006, 17(2): 126–144. [doi: 10.1287/isre.1060.0082]

[13] Valimaki M. Dual licensing in open source software industry. SSRN Electronic Journal, 2003, 8(1): 63–75. [doi: 10.2139/ssrn.1261644]

[14] Dominik R, Zo H, Maruschke M. A comparative analysis of open source software usage in Germany, Brazil, and India. In: Proc. of the 4th Int'l Conf. on Computer Sciences and Convergence Information Technology. Seoul: IEEE, 2009. 1403–1410. [doi: 10.1109/ICCIT.2009.169]

[15] Singh PV, Phelps C. Determinants of open source software license choice : A social influence perspective. Carnegie Mellon University. Journal contribution. [doi: 10.1184/R1/6705116.v1]

[16] Kechagia M, Spinellis D, Androutsellis-Theotokis S. Open source licensing across package dependencies. In: Proc. of the 14th Panhellenic Conf. on Informatics. Tripoli: IEEE, 2010. 27–32. [doi: 10.1109/PCI.2010.28]

[17] Almeida DA, Murphy GC, Wilson G, Hoye M. Do software developers understand open source licenses? In: Proc. of the IEEE/ACM 25th Int'l Conf. on Program Comprehension. Buenos Aires: IEEE, 2017. 1–11. [doi: 10.1109/ICPC.2017.7]

[18] Kapitsaki GM, Charalambous G. Find your open source license now! In: Proc. of the 23rd Asia-Pacific Software Engineering Conf. (APSEC). Hamilton: IEEE, 2016. 1–8. [doi: 10.1109/APSEC.2016.012]

[19] German DM, Manabe Y, Inoue K. The design of the questionnaire with high quality in social investigations. Automated Software Engineering. ACM. 2010. 437–446. [doi: 10.1145/1858996.1859088]

[20] Kapitsaki GM, Tselikas ND, Foukarakis IE. An insight into license tools for open source software systems. Journal of Systems and Software, 2015, 102: 72–87. [doi: 10.1016/j.jss.2014.12.050]

[21] Gacek C, Arief B. The many meanings of open source. IEEE Software, 2004, 21(1): 34–40. [doi: 10.1109/MS.2004.1259206]

[22] Skidmore D. Stakeholder value, usage, needs and obligations from differnet types of F/LOSS licenses. In: Feller J, Fitzgerald B, Scacchi W, Sillitti A, eds. Open Source Development, Adoption and Innovation. Boston: Springer, 2007. 343–348. [doi: 10.1007/978-0-387-72486-7_39]

[23] Viseur R, Robles G. First results about motivation and impact of license changes in open source projects. In: Damiani E, Frati F, Riehle D, Wasserman AI, eds. Open Source Systems: Adoption and Impact. Cham: Springer, 2015. 137–145. [doi: 10.1007/978-3-319-17837-0_13]

[24] Vendome C. A large scale study of license usage on GitHub. In: Proc. of the 37th IEEE/ACM IEEE Int'l Conf. on Software Engineering. Florence: IEEE, 2015. 772–774. [doi: 10.1109/ICSE.2015.245]

[25] Hofmann G, Riehle D, Kolassa C, Mauerer W. A dual model of open source license growth. In: Proc. of the 9th IFIP WG 2.13 Int'l Conf. on Open Source Software: Quality Verification. Koper-Capodistria: Springer, 2013. 245–256. [doi: 10.1007/978-3-642-38928-3_18]

[26] Kashima Y, Hayase Y, Yoshida N, Manabe Y, Inoue K. An investigation into the impact of software licenses on copy-and-paste reuse among OSS projects. In: Proc. of the 18th Working Conf. on Reverse Engineering. Limerick: IEEE, 2011. 28–32. [doi: 10.1109/WCRE.2011.14]

[27] Jensen C, Scacchi W. License update and migration processes in open source software projects. In: Proc. of the 7th IFIP Int'l Conf. on Open Source Systems. Salvador: Springer, 2011. 177–195. [doi: 10.1007/978-3-642-24418-6_12]

[28] Wu YH, Manabe Y, German DM, Inoue K. How are developers treating license inconsistency issues? A case study on license inconsistency evolution in FOSS projects. In: Proc. of the 13th IFIP WG 2.13 Int'l Conf. on Open Source Systems: Towards Robust Practices. Buenos Aires: Springer, 2017. 69–79. [doi: 10.1007/978-3-319-57735-7_8]

[29] Horne NT. Open source software licensing: Using copyright law to encourage free use. Georgia State University Law Review, 2001, 17(3): 863–892.

[30] Kennedy DM. A primer on open source licensing legal issues: Copyright, copyleft and copyfuture. Saint Louis University Public Law Review, 2001, 20(2): 345–378.

[31] Morin A, Urban J, Sliz P. A quick guide to software licensing for the scientist-programmer. PLoS Computational Biology, 2012, 8(7): e1002598. [doi: 10.1371/journal.pcbi.1002598]

[32] Wang XG. A comparative study of several open source protocols. Science & Technology Information, 2010, (14): 20 (in Chinese with English abstract). [doi: 10.16661/j.cnki.1672-3791.2010.14.174]

[33] Comino S, Manenti FM. Dual licensing in open source software markets. Information Economics and Policy, 2011, 23(3–4): 234–242. [doi: 10.1016/J.INFOECOPOL.2011.07.001]

[34] Hope J. Open Source Licensing. In: Krattiger A, Mahoney RT, Nelsen L, eds. Intellectual Property Management in Health and Agricultural Innovation: A Handbook of Best Practices. New York: MIHR, 2007. 107–118.

[35] Stallman RM. Free Software, Free Society: Selected Essays of Richard M. Stallman. Boston: Free Software Foundation, 2002. 91–92.

[36] Demil B, Lecocq X. Business model evolution: In search of dynamic consistency. Long Range Planning, 2010, 43(2–3): 227–246. [doi: 10.1016/j.lrp.2010.02.004]

[37] Engelfriet A. Choosing an open source license. IEEE Software, 2010, 27(1): 48–49. [doi: 10.1109/MS.2010.5]

[38] Dong HJ, Zhu DX. The design of the questionnaire with high quality in social investigations. Academic Journal of Jinyang, 2019, (5): 115–120 (in Chinese with English abstract). [doi: 10.16392/j.cnki.14-1057/c.2019.05.014]

[39] Vendome C, Linares-Vasquez M, Bavota G, Di Penta M, German D, Poshyvanyk D. License usage and changes: A large-scale study of java projects on GitHub. In: Proc. of the 23rd IEEE Int'l Conf. on Program Comprehension. Florence: IEEE, 2015. 218–228. [doi: 10.1109/ICPC.2015.32]

[40] Feng SY, Ni JX, Zou GH. Theory and Method of Sampling Survey. 2nd ed., Beijing: China Statistics Press, 2012. 32–56 (in Chinese).

[41] Cruzes DS, Dyba T. Recommended steps for thematic synthesis in software engineering. In: Proc. of 2011 Int'l Symp. on Empirical Software Engineering and Measurement. Banff: IEEE, 2011. 275–284. [doi: 10.1109/ESEM.2011.36]

[42] Smith E, Loftin R, Murphy-Hill E, Bird C, Zimmermann T. Improving developer participation rates in surveys. In: Proc. of 2013 6th Int'l Workshop on Cooperative and Human Aspects of Software Engineering. San Francisco: IEEE, 2013. 89–92. [doi: 10.1109/CHASE.2013.6614738]

[43] Duan WT, Jiang GR. A review of the theory of planned behavior. Advances in Psychological Science, 2008, 16(2): 315–320 (in Chinese with English abstract).

[44] Zhou MH, Mockus A. Who will stay in the FLOSS community? Modeling participant's Initial behavior. IEEE Trans. on Software Engineering, 2015, 41(1): 82–99. [doi: 10.1109/TSE.2014.2349496]

[45] Ajzen I. The theory of planned behaviour: Reactions and reflections. Psychology & Health, 2011, 26(9): 1113–1127. [doi: 10.1080/08870446.2011.613995]

[46] Ajzen I, Fishbein M. Attitude-behavior relations: A theoretical analysis and review of empirical research. Psychological Bulletin, 1977, 84(5): 888–918. [doi: 10.1037/0033-2909.84.5.888]

[47] Gomulkiewicz RW. De-bugging open source software licensing. University of Pittsburgh Law Review, 2002, 64(1): 75–103.

[48] Krishnamurthy S. An analysis of open source business models. In: Feller J, Fitzgerald B, Hissam S, Lakhani K, eds. Making Sense of the Bazaar: Perspectives on Open Source and Free Software. MIT Press, 2005.

[49] Holtgrewe U, Werle R. De-commodifying software? open source software between business strategy and social movement. Science Studies, 2001, 14(2): 43–65.

[50] Hall AJ. Open-source licensing and business models: Making money by giving it away. 2017. http://digitalcommons.law.scu.edu/chtlj/vol33/iss3/3

[51] Wang L. The license of 75 open source projects in 35 enterprises is analyzed. 2017. https://www.oschina.net/news/88307/75-popular-projects-open-source-licenses (in Chinese).

附中文参考文献:

[4] 达尔文. Redis模块开源许可证变更, 多个项目不再开源遭质疑. 2018. https://www.oschina.net/news/99271/redis-database-license-change

[5] 程德杰. Android的开源迷雾与中国厂商的机遇. 通信世界, 2013, (7): 11. [doi: 10.13571/j.cnki.cww.2013.07.003]

[32] 王希光. 几种开源协议的比较研究. 科技资讯, 2010, (14): 20. [doi: 10.16661/j.cnki.1672-3791.2010.14.174]

[38] 董海军, 朱东星. 社会调查中高质量问卷的设计. 晋阳学刊, 2019, (5): 115–120. [doi: 10.16392/j.cnki.14-1057/c.2019.05.014]

[40] 冯士雍, 倪加勋, 邹国华. 抽样调查理论与方法. 第2版, 北京: 中国统计出版社, 2012. 32–56.

[43] 段文婷, 江光荣. 计划行为理论述评. 心理科学进展, 2008, 16(2): 315–320.

[51] 王练. 分析了35家企业75个开源项目的许可证. 2017. https://www.oschina.net/news/88307/75-popular-projects-open-source-licenses

吴欣(1990－), 女, 硕士生, CCF 学生会员, 主要研究领域为开源生态模式与机制.

王志强 (1995－), 男, 硕士, CCF 学生会员, 主要研究领域为开源软件, 计算机视觉.

武健宇 (1997－), 男, 博士生, CCF 学生会员, 主要研究领域为软件仓库挖掘, 开源软件生态系统.

杨丽蕴 (1981－), 女, 高级工程师, CCF 高级会员, 主要研究领域为云计算, 开源, 应用软件, 标准化.

周明辉 (1974－), 女, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为软件仓库挖掘, 开源软件生态系统.