

In partnership with:

 **LF** AI & DATA

将内部代码发布为一个新的开源项目：干系人指南

2022年11月

Ibrahim Haddad 博士
LF AI & Data 和 PyTorch 基金会 执行董事

目录

| | |
|--------------------------|----|
| 信息图..... | 4 |
| 摘要 | 5 |
| 概述 | 6 |
| 前期调查 | 8 |
| 开源商业方案 | 8 |
| 评估可能的开源方式..... | 8 |
| 项目资金 | 9 |
| 法律考虑..... | 9 |
| 确认代码所有权..... | 9 |
| 进行知识产权审查..... | 10 |
| 选择开源许可证 | 10 |
| 应用许可证条款到代码 | 11 |
| 清理代码..... | 11 |
| 项目品牌 | 12 |
| 制定商标战略和政策..... | 12 |
| 域名 | 12 |
| 创意资产..... | 12 |
| 注册外部账户..... | 12 |
| 制定认证 / 合规策略..... | 12 |
| 招募商业伙伴 | 12 |
| 建立项目治理制度..... | 13 |
| 建立项目基础设施..... | 14 |
| 为你的 GitHub 仓库应用推荐做法..... | 14 |

项目启动 15

 准备公告..... 15

 媒体和分析师关系 15

 宣布并启动项目..... 15

运营一个开源项目的实践经验..... 16

 开源协议使用 16

 治理模型..... 16

 使用权..... 16

 流程 16

 项目发展 16

 社区运营..... 16

 社区结构..... 16

 版本发布..... 17

 沟通工具..... 17

 透明度..... 17

 文档工作..... 17

持续支持..... 18

 支持社区..... 18

 支持项目基础设施..... 18

尾注 18

根据 **OSI 批准的开放源代码许可证** 来许可你的项目以自由创建和分发衍生品。



建立一个透明平等的治理模式，以**支持项目健康、长期发展和增长。**

允许任何有兴趣参与项目且做出贡献的人**公开访问项目资源。**



记录所有项目过程，以维护有关提交，请求，同行评审和成员角色的相关标准，并对基于社区反馈的修订保持开放。



og



在进入最终版本前，通过**满足质量标准的合格人员进行多级审校**来管理项目开发

建设社区文化，努力实现**可访问性、可见性、自组织性和弹性。**



构建社区来为**贡献价值**的人推广扩展活动，无论他们是新人，固定维护人员，最终用户还是开发者。



以**明确和透明的节奏**稳定发布，以推广新功能，同时为用户和开发人员维持可靠性和安全性。

为任何希望参与项目的人提供**开放和可访问的沟通工具。**



通过召开会议和活动，发布项目更新，回答问题，提交补丁以及创建可度量可追踪的 **KPI 目标**，**定期提供社区支持。**



为了吸引参与者，在贡献，同行评审，讨论和维护者或提交者宣传中**保持透明度。**



公开所有文档，包括架构，API，教程以及安装，开发和参与指南。

摘要

随着企业意识到消费和贡献开源项目的价值, 其在开源方面的参与度已达历史新高, 且持续增长。企业越来越多地发现, 开源自有技术可以创造新的价值源和更强大的产品生态, 其参与的性质在不断演进。

开源一种自有技术不只是局限在源代码开放, 有很多方式来建立或加入社区, 以应用并帮助维护项目, 因此它应当是一个条理清晰且深思熟虑的过程。

对于有计划开源自有代码成为一个独立开源项目的企业, 本文给出了该过程的顶层概述, 并提供了一个示例的检查清单, 有助于确保正确识别和执行所有任务。

概述

开源软件 (OSS) 一直在并将持续推动软件行业发展到一种新范式, 从闭门开发专有代码, 转向为开发各方可以公开共享、修改和重新分发的代码。这种转变的主要好处在于降低开发成本和软件组件复杂性, 开发可复用的通用标准软件资产, 提高灵活性, 并受益于社区驱动开发项目带来的多重创新。组织将开源模式作为构建软件的积极手段, 可以增加其保持竞争优势的机会。图 1 说明了开源软件为采纳和促进开源软件的组织所提供的各种战略优势。

(OSPO) 成为一种趋势, 组织通过它来管理开源软件的各个方面, 包括开源软件许可证的使用和遵守、对开源软件项目的贡献以及围绕开源软件关键技术的社区建设。

在过去的二十年中, 企业组织已经意识到在其产品和服务中采纳和贡献开源项目的好处。这导致设立开源项目办公室

图 1
为何开源?



图 2

企业开源阶梯

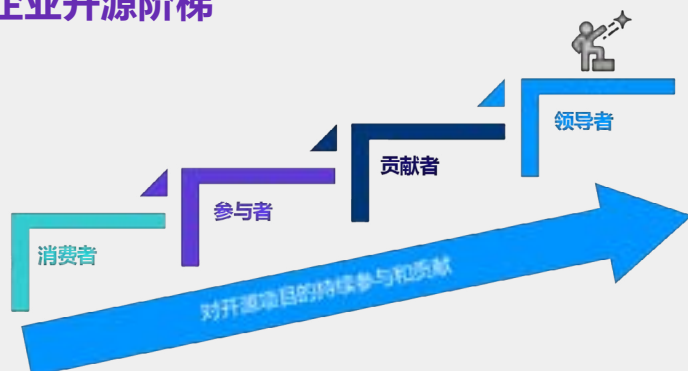


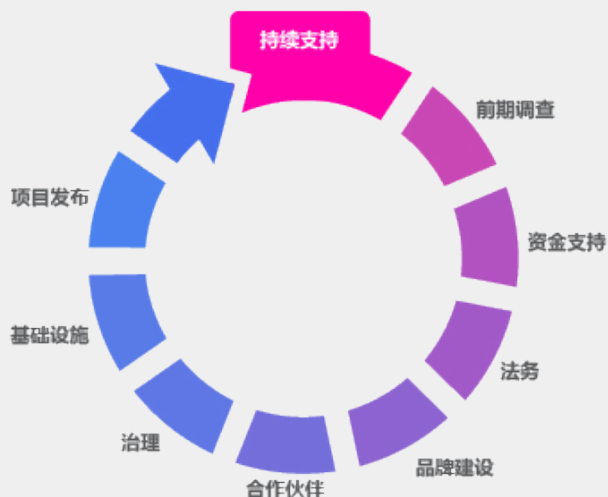
图 2 说明了四种主要的 OSS 企业战略：OSS 消费、参与、贡献和领导。每一项战略都要求企业在前一项战略中取得成功，而组织的进步程度完全取决于企业。工程推动了消费和参与的早期战略。工程师使用各种开源组件以提高开发速度，但他们很少参与维护这些组件的项目。随着时间的推移，组织的高层了解到 OSS 使用的价值。随着 OSS 的发展，业务需求开始推动 OSS 的参与，OSS 的努力有助于确定业务战略。一些公司实现了作为消费者的目标。其他公司在参与的其他阶段看到了战略优势，在大多数情况下，他们设立了一个 OSPO 来监督这些阶段的战略规划和执行。

作为第三阶段——贡献开源——的一部分，组织通常会以各种动机为开源贡献关键专有技术，例如：

- 提供标准的参考实现
- 确保关键软件持续可用
- 削弱竞争对手
- 开辟商业化市场
- 与他人合作并提升在开发者社区的信誉
- 准确把握市场需求，构建上下游生态
- 支持客户自助服务及增加自定义功能

图 3

开源过程的相关步骤



带有错误动机的开源通常会对实现预期结果产生负面影响，并可能破坏企业与特定开源项目社区的关系。

本文确定了在开源专有技术时要明确的问题、要考虑的实践以及要采取的具体步骤。图 3 说明了将内部代码发布为一个开源项目过程中涉及的各个步骤。这些步骤不一定按顺序逐个执行，其中一些步骤可以并行进行。本文的目标是提供一个基本模板，组织可以根据自己的政策和战略按需调整。

前期调查

在开源自有技术时, 必须彻底评估转型的原因, 并相应地调整内部激励和相应指标。出于错误的动机进行开源可能会产生与最初预期相反的效果。想要成功地开源项目, 你必须得有正确的理由或动机。

开源商业方案

有许多合理的商业理由来开源自有代码, 比如:

- 强化正在开发中的产品或服务的生态系统
- 吸引业务合作伙伴和客户参与增强功能和修复缺陷, 以提升产品质量
- 为某个标准提供参考实现, 从而推动该软件成为事实上的标准实现
- 推动软件栈的非战略层面商业化
- 推动更高价值和更多创新
- 与开源社区合作, 提升在开发者社区的好感度

同样地, 开源自有代码也有许多因素会导致事与愿违。以下这些应当作为危险信号:

- 希望有其他人来维护你依然需要的代码库, 以便你自己可以停止对该代码的持续投入
- 希望淘汰那些你不打算再维护或使用的独特功能的代码
- 源代码直接链接到了你不能在开源许可证下发布的代码

你现在已经有了在组织中开源代码的商业案例, 下一步就是要确定具体的开源方式。

评估可能的开源方式

并不存在什么唯一的方法来达成可能的目标, 并且这也不是要你的组织必须单独完成的。在大多数情况下, 你可以探索多种可能性, 例如:

- 做好技术评估并确定是否应该同时开源其他组件
- 分析那些适合你的公司作为主要参与者加入的已有开源项目, 尽量避免创建新的基础设施和新的社区
- 探索与客户和合作伙伴一起发布计划中的开源项目的可能性
- 评估有发布并维护成功的开源项目的开源基金会, 通过其发布和托管计划中的开源项目。

项目资金

商业方案一旦开源, 你就需要制定一个项目计划以及各阶段的预算, 包括随着时间的推移启动和维护项目的成本; 这些费用中的一部分是一次性的, 而其他的则可能反复出现。举例来说, 这类成本可能包括:

- 通过有计划的内部法务活动指引公开的代码发布
- 持续的 IT 基础设施和云服务配额 (如适用) 支持
- 商标管理
- 创意及品牌设计 (徽标、网站、活动标识等)
- 项目管理
- 需要定期执行的开源许可证合规性和安全性扫描
- 社区活动与黑客马拉松

法律考量

在法律层面, 有 5 个主要的活动需要执行, 包括:

1. 确认开源项目相关的所有源代码的所有权
2. 进行知识产权审查
3. 选择一个开源许可证
4. 应用许可证条款到代码并更新源代码中的许可证信息
5. 在公开发布前清理源代码

在下面的小节中, 我们将介绍这些活动, 并提供适当的建议。

确认代码所有权

开源自有技术的风险之一是意外将第三方专有代码作为开源代码的一部分。在根据开源许可证发布任何代码之前, 强烈建议组织确认拥有开源代码所需的所有权利和权限。

可以按照以下步骤方式执行:

- 使用软件组成分析工具(SCA)审计源代码¹
- 识别第三方代码, 确定是开源的还是商业的
- 确认你是否有权根据开源许可证对任何发现的第三方商业代码进行开源; 如果答案是否定的, 则无法开源那些第三方代码, 你需要提供可替代代码

进行知识产权审查

可能受到专利或其他知识产权保护的软件, 不是开源发布的理想候选软件。回答以下问题可以帮助你完成此活动:

- 该代码是否披露或实现了公司计划通过专利保护的任何发明?
 - 如果答案是肯定的, 那么需要决定是移除代码、建立 IP 策略还是做出不声明承诺。在任何情况下, 一旦出现这种情形, 你的法律顾问都会为接下来的步骤提出适当的建议。
- 该源代码的发布会引发对开源软件的专利索赔吗?
 - 如果答案是肯定的, 那么你必须移除并替换受保护的代码, 或寻求更适合的许可证或权限。
- 你为开源项目所选的名称 (假设您正启动一个新项目) 是否受商标法保护? 与项目相关的或已注册的名称或商标是否存在侵权索赔风险?

选择开源许可证

开源项目的许可证定义了使用、复制、修改和分发代码的权利。开源项目的许可证选择是决定项目开放性的一个重要因素。开源项目只能使用 [Open Source Initiative](#) 已批准的许可证。这类许可证允许软件自由使用、修改和共享。要获得 Open Source Initiative 的批准, 许可证必须通过 [license review process](#) 确认其满足 [Open Source Definition](#) (OSD)。你可能会遇到许多其他与 OSD 不兼容的许可证。这些许可证中的大多数是 “Source Available” 许可证, 通常包括对软件的使用和/或分发的约束或限制。这些限制通常使许可证与 OSD 不兼容。

始终建议采用 OSI 批准的开源许可证。具体许可证的选择取决于你作为一个组织想要实现的具体目标。在本节中, 我

们将预览一些问题, 这些问题将推动关于此主题的讨论, 并帮助你决定所要采用的许可证。

- 是否要放弃控制他人如何使用和分发代码?
- 是否允许他人在商业程序和产品中使用该代码?
- 如果他人使用其程序中使用代码并将其出售, 你想要获得这些收入的一部分吗?
- 如果他人使用和分发代码并对其进行改进 (例如修复漏洞或增加特性), 你是否希望他们将这些改进贡献给项目?
你希望在何种许可证下接受贡献代码?
- 是否需要开发者溯源认证 (DCO) 或贡献者许可协议 (CLA) 作为贡献要求?

作为此活动的一部分, 可能还需要考虑其他问题, 这主要由你的法律顾问和组织内的技术领导者推动。

DCO

DCO 签署流程确保了项目中接受的每一行代码都有清晰的审计跟踪。这是开发者有权提交代码到项目中的凭证。例如, Linux 内核流程要求所有贡献者为其提交的代码做签名, 以示贡献者证明自己的代码如 [DCO](#) 所述。签名表明贡献者根据适当的开源许可证创建或接收了贡献, 该许可证根据文件中所显示的许可证将其纳入项目的代码库。DCO 建立了一个负责许可和证明项目贡献的人员链。

CLA

有些项目要求开发者或其雇主签署 CLA。与 DCO 不同, CLA 的内容可能会因项目而异, 因此任何给定的 CLA 条款可能具有不同的效果。CLA 的目的是确保项目产出的监护人对所有贡献拥有必要的所有权或授予的权利, 以允许他们在选定的许可证下进行分发。在某些情况下, 这甚至意味着贡献者将授予不可撤销的许可, 允许项目将贡献作为项目的一部分进行分发。

应用许可证条款到代码

一旦按照上一步中提供的建议清理了源代码, 就应该将许可条款应用于代码。该项活动包括以下步骤:

- 在包含完整许可文本的组件的根目录中添加许可文件。例如, 如果你在 GitHub 上发布代码, 就要提供一个

LICENSE.md 文件, 其中包含开源许可证的全文。

- 向每个源代码文件添加许可证头信息和版权声明
- 在项目网站上明确指定许可证, 常见问题 (FAQ) 以及下载页面 (如果适用)
- 在源文件中使用 SPDX2 许可证列表 “短标识符”

清理代码

开源一个项目的另一种风险是, 包含了私有信息、通信凭证以及商业密钥等。为了最小化此类风险, 可以参考下列行动项:

- 移除暴露的任何非公开应用程序接口 (API)。
- 删除任何包含员工姓名或个人识别信息, 产品代号, 路线图, 未来产品描述或贬损的评论。
- 从源代码中删除任何未使用或过时的代码, 以增加社区做出贡献的可能性。
- 创建并包含一个包含所有第三方软件的许可证和版权声明的文件, 并在适用时提供源代码。
- 如果源代码有第三方代码的依赖, 则需向社区提供必要的信息; 最好避免代码对非开源组件的依赖。
- 删除所有第三方私有代码。

项目品牌

项目品牌建设包括一些应该考虑的活动, 我们在下面的小节中讨论。

制定商标战略和政策

- 约定项目的名称 / 标志。
- 通过淘汰式搜索确定名字是否被占用。
- 指定商标的内部联系人 (该联系人不能是律师)。
- 制定注册策略:
 - 适用于哪些类别的商品/服务?
 - 哪些国家要优先考虑?
- 注册商号和商标。

域名

- 注册域名并设置重定向

创意资产

- 设计 logo、logo 包装和视觉资产。
- 设计并在网站上发布 logo 使用指南。

注册外部账号

- 在 GitHub 上创建项目组织名。
- 在不同社交媒体平台如 Twitter、LinkedIn 和 Facebook 上创建 @项目名称 账号。

制定一个认证 / 合规战略

- 制定合规标准, 如果某项目对外声明与母项目兼容, 需遵循此标准。

- 创建规范文件和可验证项目的定制构建是否符合规范的工具。
- 约定认证项目的名称/标志。
- 如果你想控制项目名称的使用, 就制定一个商标政策/常见问题列表。提出下面这些问题, 以引导关于该主题的对话:
 - 贡献者, 用户组或开发者是否可以注册含有项目名称的域名?
 - 该项目是否会实施认证程序, 以允许其他人使用修改后的商标?
- 创建认证测试套件。
- 与测试机构建立契约。
- 安排第一年的互操作性测试。

招募商业伙伴

- 与将从项目中受益最多的商业伙伴接触, 以便在启动日获得公开支持。
- 确保得到主要合作伙伴的承诺, 鼓励他们的员工参与项目并提供一些基础的贡献。
- 接洽兼容的项目, 沟通新项目将如何使他们受益, 并让他们为公告做好准备。
- 预测哪些现有的项目会误认为此项目与他们有竞争关系, 并提前化解。
- 让商业伙伴尽早接触到项目源代码。
- 与合作伙伴合作, 为共享客户建立一个联合价值主张和参考堆栈。

建立项目管理制度

治理制度决定了谁对项目的影响和控制超出了开源许可中的基本要求。一个项目的管理制度建立了一套解决以下疑难问题的协作框架, 如:

- **项目贡献**
 - 谁来决定代码的收录和发布, 以及如何决定?
 - 谁是项目的主要维护者或架构师 (较大的项目有不止一个)?
 - 项目的贡献者如何成为维护者或提交者?
- **方向和资金**
 - 项目如何筹备资金, 谁可以决定资金的用途?
 - 该项目是否应该有一个技术指导委员会 (TSC) 或合规性认证委员会? 谁可以加入它们?
 - 谁来决定项目的方向和路线?
- **透明度**
 - 谁能参与讨论并决定关键事项?
 - 决策过程的透明度如何?
 - 谁能参与项目中发生的讨论和会议?
- **复用**
 - 对重新分发、修改或使用软件有什么合规要求?
 - 项目如何使贡献者和下游的再分配者遵守这些要求?
- **版权和商标**
 - 谁拥有贡献代码的版权?
 - 用户如何许可项目的品牌?

通常情况下, 项目的初始维护者构成项目的 TSC。这些人通常来自项目的创始组织。我们的目标是随着时间的推移增加 TSC 的成员, 吸引高价值的贡献者。

- **项目管理制度**
 - 确定 TSC 的成员。

- 确定 TSC 的主要职责, 如:
 - » 监督软件架构和实施活动
 - » 起草发布计划和路线
 - » 与项目所依赖的其他开源项目合作
 - » 制定接受 / 拒绝代码的标准
 - » 处理源代码安全问题
- **项目流程:** 一个高度开放的项目会清楚地定义社区工作事务流转以及如何为项目做出贡献的流程。对于新手来说, 一个清晰的开发流程应该展示如何将代码合入项目、发布流程和项目的计划表、开发者需要满足的要求, 还应该包含诸如补丁提交、功能请求, bug 报告以及代码贡献签名等的社区最佳实践的参与准则。
 - 功能请求
 - 发布管理
 - 代码提交
 - bug 报告
- **项目协议**
 - 制定一个第三方贡献协议, 以治理项目如何管理来自社区的贡献。

建立项目基础设施

- 文档
- 项目网站
- Wiki
- 社区交流频道
 - 邮件列表
 - 即时通讯 (如 Slack, 互联网中继聊天 (IRC))
- 协作平台
 - Wiki
 - GitHub 仓库 (或者自己搭建 git 服务器)
- 缺陷追踪和功能请求
- 构建系统

推荐你的 GitHub 仓库实行的措施

1. 使用通过 [TODO Group](#) 创建的 [REPOLINTER](#) 工具识别你仓库的通用问题。
2. 通过 [双重身份认证](#) 提高 GitHub 账号的安全性。
3. 每个仓库包含一个 **LICENSE** 文件。
4. 在你的仓库中添加一个 **README** 文件, 欢迎新的社区成员, 阐述该项目的重要性, 介绍开始方法。
5. 在你的仓库中添加一个 **CONTRIBUTING** 文件, 向其他开发者和你的用户社区介绍如何为项目做贡献。更进一步, 你也可以说明项目需要什么类型的贡献, 以及如何贡献。
6. 添加一个 **CODEOWNERS** 文件来说明谁负责版本库中的代码。

7. 添加一个 **CODE_OF_CONDUCT** 文件, 为参与者的行为制定基本规则, 并帮助促进一个友好, 欢迎新人的环境。虽然不是每个项目都有 **CODE_OF_CONDUCT** 文件, 但它的存在表明这是一个欢迎贡献的项目, 并定义了参与项目社区的标准。
8. 提供关于发布方法, 节奏, 标准等等的文件。
9. 记录你的项目管理, 并将其放在项目仓库里。
10. 添加一个 **SUPPORT** 文件, 让用户和开发者知道如何获得你项目的帮助。你可以在项目根目录下添加一个 **README** 说明如何处理安全问题, 也可以链接到安全文档。
11. 设置一个 issue 模板和 pull request 模板, 帮助你定制和规范你希望贡献者在你的仓库中打开 issue 和 pull request 时包含的信息。
12. 实现并保持你的项目的 [OpenSSF 最佳实践徽章](#)。(以前称为核心基础设施倡议最佳实践徽章)。
13. 确定谁来处理安全问题 (可能是一个团队), 并且建立一个单独的电子邮件账户。
14. 考虑让该项目成为 CNA (CVE编号机构)。
15. 在合理的情况下, 在仓库中每个文件的顶部的注释中包括一个 SPDX 的短式标识符。
16. 采用 [GitHub DCO app](#), 在每次提交中强制使用 "Signed off-by: " 标签。DCO 是一种简单的方法, 可以让贡献者证明他们写了或有权利提交他们所贡献的代码。该应用程序在 pull request 上执行 DCO。它要求所有的提交信息必须包含 **Signed-off-by** 行, 并带有与提交作者匹配的电子邮件地址。
17. 在 GitHub 上发布的任何内容都要使用英语作为默认的通用语言。你可以支持第二种语言, 但英语应该是与全球受众交流的主要语言。

项目启动

公告的准备

- 简要介绍合作伙伴
- 检查所有项目基础设施是否正在运行, 安全且可扩展。
- 为关键项目人员订阅和配置项目邮件列表。
- 确保内部开发人员加入并持续对内部沟通进行监控。

媒体以及舆情监测

- 制定启动策略和时间表。
- 起草新闻稿并征得所有相关方的同意。
- 确定发言人和媒体联系人。

- 创建内部和外部常见问题解答 (FAQ)。
- 对媒体和舆情进行持续监测。
- 制定持续的公共关系/舆情分析战略。
- 如果需要, 外部聘请 PR/AR 公司以全面实施该战略。

宣布并启动项目

- 发布源代码。
- 发布一份路线图, 即使它是不成熟的设想。
- 遵循开源开发模式。
- 综合监控和分析 PR/AR 策略带来的效果。

运营一个开源项目的实践经验

开源协议的使用

通过 OSI 批准的开源许可证赋予项目创建和发行衍生产品的自由。

治理模型

一种为项目的所有当前和未来贡献者提供平等地位的治理模型。具有开放透明治理模型的开源项目有更好的发展机会, 以及健康发展的生态, 并会吸引更多的开发人员和采用者。

使用权

对项目感兴趣的任何用户或开发人员都可以访问项目资源。任何人都可以参与该项目, 任何参与者都可以通过贡献和与项目社区建立信任关系来获得提交者的权利。

流程

- 一般来说项目中比如提出功能需求, 为项目提交错误报告, 提交代码等都会被文档记录。
- 源代码的贡献需要通过达成共识的提交流程进行。
- 所有代码都经过同行评审过程。
- 由于项目需要确保一致性, 成为提交者 / 维护者 / 审查者需要严格遵守流程。
- 项目社区根据反馈修改其流程, 以确保随着项目的发展和规模变化不断满足和适应项目的需求。

项目的发展

- 发展的责任需要分配给具有最佳交付能力的个人。

- 项目在合并代码时需要严格执行质量标准。
- 项目在发布版本前需要进行多级审核。
- 同行评审是强制性的和公开的。

社区的运营

- 对新手友好 —— 项目通常都是开放式开发。
- 社区注重开放的决策过程和沟通方式, 突出项目的治理透明和可视化。
- 良性的社区是自驱型组织; 项目中的参与者, 无论是个人还是组织都积极的在其关注和擅长的领域对项目做出贡献。
- 由于领导力来自经验和演进, 因此参与项目需要能够适应组织的变化。如果有开发者停止参与, 也会有别的开发者代替他们。

社区结构

- 采用精英管理推动社区的形成。那些为社区提供最大贡献的参与者会被社区选为项目的领导角色。
- 项目欢迎积极参与公共讨论、开发和测试的贡献者加入。
- 项目的层次结构是可扩展的, 负责监督不同代码区块的维护者组成了代码维护团队, 随着社区的规模的变化, 这个维护管理的层级可以根据需要增加或减少。
- 任何人都可以提交补丁, 开发人员和用户都参与了测试过程。开发者和用户的角色与开源开发紧密结合, 让用户有更直接的途径来影响项目。

版本发布

- 项目在不断的迭代, 为了让某些用户避免使用不稳定的开发中版本, 项目会将部分新的实验特性进行有限制的发布, 确保用户在稳定版本中使用已经有过案例测试过的功能。
- 每周或每月为用户和开发者提供经过测试后的最新功能的稳定版本。
- 长期稳定版本扩展到更长的发布周期, 而且通常只包括安全补丁和错误修复。
- 项目需要有明确的发布节奏和每个版本的既定目标。
- 发布节奏和每次发布要达到的目标是所有项目利益相关者都需要周知的。

沟通工具

包括邮件列表、Slack 和 IRC 等在内的有效工具都可以被项目选用, 并开放给任何希望参与该项目的人。

透明度

开源社区必须尽可能透明以吸引新的人参与, 例如贡献透明度, 同行评审透明度, 讨论透明度以及向提交者或维护者晋升的透明度。

文档工作

文档是社区的重要组成, 需要提供涵盖体系结构, API, 安装指南, 开发人员指南, 开发过程, 参与指南, 教程等的文档, 便于新用户了解项目, 开发者参与项目。

持续支持

项目启动之后, 应对外部社区的活跃度进行持续跟踪, 并从各个领域对项目提供支持, 从而对其进行扶植并支持相关社区的成长。

对社区的支持

- 定期与主要干系人沟通。
- 通过网站、公关和社交媒体定期发布项目更新信息。
- 通过沟通渠道回复来自社区的提问。
- 对提交的补丁进行审查并在必要时拉入代码库。
- 通过活动协调, 实现社区培育和技术推广。
- 为项目成功量身定制一套KPI, 跟踪这些指标, 制定计划并落实, 以确保实现这些目标。

对项目基础设施的支持

- 确保网站和 Wiki 的实时更新。
- 及时为商标律师提供指导。
- 常态化的商标管理。
- 域名注册和续订管理。
- 交流平台的监督与引导(邮件列表, IRC, 论坛等)。
- 媒体、评论员的关系维护。
- 制定可持续的 PR/AR 战略, 并通过与企业合作, 谋求相应的服务支持。

尾注

- 1 SCA工具是软件开发团队用来检测开源许可证合规性并提高代码安全性的一系列应用程序。高级的SCA工具可以对源代码库进行自动扫描。它们还可以帮助团队识别开源代码组件及其许可证, 并对已知的安全漏洞进行标记。
- 2 软件包数据交换® (SPDX®) 是一项可用于组织之间或上游开源项目到组织间传递的软件物料清单开放标准。

总结

开源专有技术成功的途径多种多样。本文对这个流程提供了一种高级概括,可作为进一步制定详细内部计划的基础。需要强调的是,应当意识到这个清单可能并不完整,并且在不同组织和项目之间也有差异。其目标是提供与内部开源项目相关的最普遍任务,并帮助它们简化流程。虽然这个过程看起来可能复杂又漫长,但许多组织已经通过遵循类似过程,成功将内部代码作为开源项目推向市场。同时,这个过程也将很多任务环节自动化,并使用项目管理工具来协调和跟踪所有任务。

了解更多关于创建成功开源项目以及与开源社区合作的信息,请访问 Linux 基金会网站,在这里您将获得大量免费资源,助您开启开源之旅。

鸣谢

作者对Linux基金会同事 Hilary Carter、Jason H. Perlow 和 Melissa Schmidt 表示衷心的感谢,感谢他们的宝贵点评和反馈意见。这份报告极大地得益于他们的经验和贡献。

Linux基金会资源

- 电子书:《[带你深入开源项目办公室](#)》
- 电子书:《[企业开源指南](#)》
- 电子书:《[企业开源合规性](#)》
- 电子书:《[并购交易中的开源审计](#)》
- 《[Linux基金会企业指南](#)》
- [Linux基金会开源规范](#)—组织开放源码合规支持资源
- [TODO工作组](#)—一个由从业人员和组织组成的开放社区,他们通过最佳实践、工具和其他方式进行协作,以成功运行开源程序。
- [软件包数据交换 \(SPDX®\)](#)

意见反馈

作者为任何拼写错误或其他可能存在的错误提前表示歉意, 并对通过 ibrahimatlinux.com/contact.html 收到的更正和改进建议表示感谢。

作者简介



Ibrahim Haddad 博士是 Linux 基金会战略项目副总裁。他致力于为开发人员提供一个集编码、管理和扩展为一体的开源项目中心, 从而打造一个与供给侧无关的环境来推进开源 AI 平台, 进而为几代开源创新者提供支持。Haddad 领导着 LF 人工智能与数据基金会和 PyTorch 基金会。他及这两个基金会都在为支持公司、开发人员和开源社区开展的解决行业与技术挑战、造福所有参与者的技术项目的识别和贡献而努力。

推特: [@IbrahimAtLinux](https://twitter.com/IbrahimAtLinux)

网址: IbrahimAtLinux.com

有趣项目: [Tux NFT Club](#)

免责声明

本报告是“以现状提供”的。Linux 基金会及其作者、贡献者和赞助商明确否认任何承诺 (明示、暗示或其他), 包括与本报告相关的可销性、不侵权、适合特定目的或标题的暗示性承诺。在任何情况下, Linux 基金会及其作者、贡献者和赞助商都不对任何其他方的利润损失或任何形式的间接的、特殊的、偶然的或任何性质的后果性损失负责, 无论是基于违反合同、侵权行为 (包括过失), 还是其他原因, 以及他们是否被告知这种侵害的可能性。赞助编写本报告并不代表任何发起者对其调查结果的认可。

感谢以下Linux 基金会 APAC 开源布道者翻译 SIG 的成员, 为本《将内部代码发布为一个新的开源项目: 干系人指南》翻译成简体中文作出了贡献。该团队成员包括:

1. 赵振华
2. 杨振涛
3. 李明康
4. 赵梅
5. 皮冰峰
6. 王誓伟
7. 刘晓斌
8. 腾召智
9. 周黎
10. 徐斌
11. Donald Liu, Linux Foundation APAC
12. Maggie Cheung, Linux Foundation APAC



twitter.com/linuxfoundation



facebook.com/TheLinuxFoundation



linkedin.com/company/the-linux-foundation



youtube.com/user/TheLinuxFoundation

LF AI & DATA

做为 Linux 基金会的一部分, LF AI & Data 支持人工智能、机器学习、深度学习和数据方面的开源创新。LF AI & Data 的建立是为了支持一个可持续的开源人工智能生态系统, 使其能够轻松地使用开源技术创建人工智能和数据产品和服务。我们在一个中立的环境下促进协作, 以开放的治理模式支持开源技术项目的调和与加速。



Linux 基金会研究部成立于2021年, 旨在研究日益扩大的开源协作规模, 提供对于新兴技术趋势、最佳实践和开源项目全球化影响的洞察。基于对项目数据库与网络的利用, 并承诺使用定量和定性方法的最佳实践, Linux 基金会研究部正在打造开源见解的首选资料库, 造福于世界各地的组织机构。



版权所有 © 2022 [Linux 基金会](#)

本报告采用[知识共享署名4.0国际许可协议](#) (CC4.0) 进行许可。

如需了解详细工作内容, 请参考原文: “开源中的人工智能和数据: 大规模集同的挑战和机遇”, 作者: Ibrahim Haddad, Seth Dobrin 博士 (序), 2022年3月。