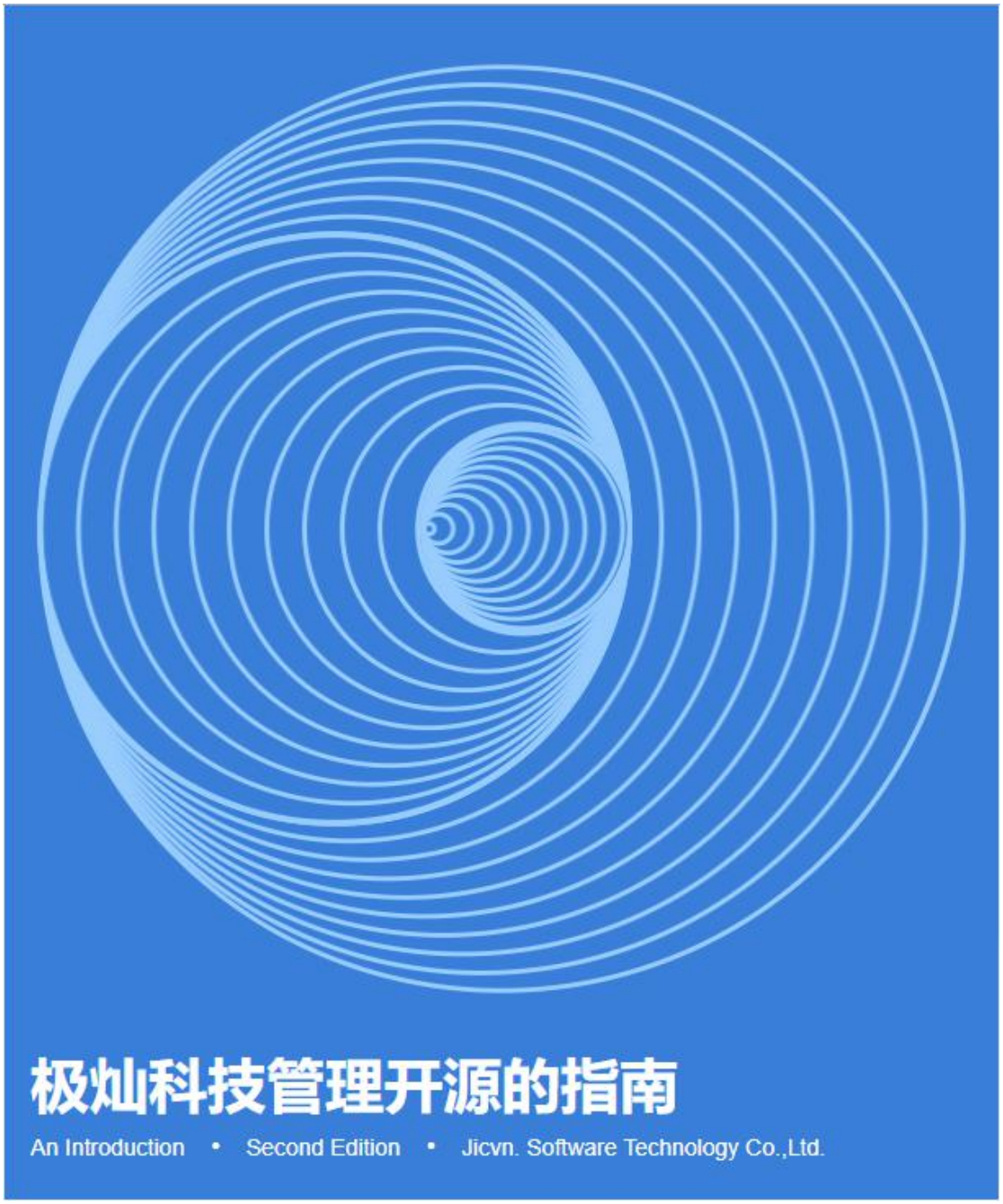


极灿科技管理开源的指南

An Introduction • Second Edition • Jicvn. Software Technology Co.,Ltd.

极灿科技指南
---开源管理

软件开发团队寻求的最佳实践
优化其开源组件的使用

导言

开源无处不在

今天，开源构成了大多数专业软件项目的基础。极灿科技最近对专业开发人员的研究显示，平均而言，他们项目中 92％包含开源组件。68％的受访者表示他们所有的项目均包含开源组件。

这也就不足为奇了。 开源为创新者提供了巨大的先机。开发人员可以使用数十亿行现有代码开发，开发由开放的合作者社区共享。

我们的意思是，它确实无处不在。

许多人在独立系统级软件的环境中了解开源，像 Linux，或 Apache Web 服务器或 MongoDB 数据库之类的应用程序。多少团队没有意识到他们内部团队开发的应用程序也是主要使用开放源码库和组件构建。

如果不是大多数，很可能至少有一些，底层代码应用程序来自开源项目。

开放源码的普及在许多方面对专业开发人员有利。

但是对开源组件的依赖也带来了一些独特的挑战。

您是否可以轻松了解这些安全性，许可和维护基本构建块？

您的团队是否能够找到确保开源的时间您使用的组件是否得到妥善维护？

这些问题不一定是新问题。商业开源服务像 Red Hat，Cloudera 和 MongoDB 这样的公司已经存在了几十年。重要的是，这些订阅并没有覆盖绝大多数您的业务所依赖的开放源码库。

堆栈向上移动到应用程序层，就是开源公式开始变得有趣的时候。在这里，开发人员从 37 种不同语言的包管理器中选择了超过 330 万个开源组件。很明显，适用于 Linux 和其他系统级开源项目的“发行版”模型无法扩展到覆盖应用程序级别的这一庞大的开源软件。

商业开源服务的有限范围

开源的隐性成本

由于大多数应用程序级开放源码软件都没有商业产品，专业开发团队在维护方面就只能束手无策，包括集成、安全和许可审查。

更糟糕的是，只有一小部分企业系统地处理这些问题。最近的调查告诉我们，只有 9%的开发团队拥有引入新依赖关系的正式流程。几乎有四分之一的开发团队在单个开发人员级别做出这些决策。这使团队面临巨大的风险和主要的生产力下降。当您考虑以下成本时，免费软件就更少了：

成本# 1:

紧跟

开发团队为了跟上不断发展的开放源码依赖关系而疲于奔命。将宝贵的时间浪费在组装、集成和测试组件上，以确保它们能够很好地协同工作。

成本# 2:

保持安全
开发团队还承担着监控其集成的数百个单独组件的安全状态的重大责任，或者承担类似影响 Equifax 的入侵风险(还有无数类似的恐怖故事)。
成本# 3:
保持兼容
所有这些优秀的开源组件都带有一系列令人困惑的不同类型的开源许可。有时很难知道您团队使用的是否符合许可条款—更不用说不同组件的许可彼此之间的配合程度了。

让您的团队回到发货代码
所有这些都提出了一个非常简单的问题……如果专业开发团队没有在上面这些任务上花费太多的时间，他们还能实现什么呢?
如果他们有更好的方法来确保开源组件能够很好地协同工作，他们可以实现什么？如果他们不必主动花时间 - 或更糟，反应性地解决安全问题。 如果他们不必担心许可问题，或者他们想要使用的软件是否完全获得许可？
在接下来的几页中，我们将分享一个关于在专业开发环境中使用开源软件的新思路的完整愿景。这种方法可以显著地影响组织从其对开源的投资中看到的结果。

本指南的目标：
探索专业开发团队在使用开源软件构建应用程序时所面临的问题
展示极灿科技如何通过以下三个步骤轻松实现主动管理您的组织开源使用的方法
识别
帮助您理解当前的依赖关系和基础，维护、安全和许可问题
事故预防
在问题出现之前帮助你解决
决议
建立一个托管的开源程序，以长期保护您的组织
第一步:识别
理解开源软件的风险领域和机会
什么是包管理?
开源软件组件或库由称为包管理器的工具管理。每个编程语言生态系统或“堆栈”通常都有一个不同的包管理器。
例如，JavaScript/Node.js 生态系统主要使用 npm 包管理器，Java 生态系统主要使用 Maven, Python 生态系统通常使用 Python 包索引(PyPI)。
开源软件开发通常在托管的源代码协作平台上进行，比如 GitHub、GitLab 和 Bitbucket。然后将其组装到包发行版中，这些发行版将进入公共包管理器存储库。为了节省时间，并使确保软件安装和正确运行变得更容易，开源的大多数商业用户从包管理器存储库获得他们的软件，而不是直接从这些代码协作平台获得。
什么是包管理器?
软件包管理器是一组软件工具，它以一致的方式自动化安装、升级、配置和删除计算机程序的过程。软件包管理器可帮助确保软件正常运行。

极灿科技有什么帮助?
➜ 极灿科技为您提供了几乎所有编程语言和开源社区中这些包管理器和包的全面视图
➜极灿科技已经构建了世界上最完整的开源软件包索引（包含超过 330 万个软件包的许可和

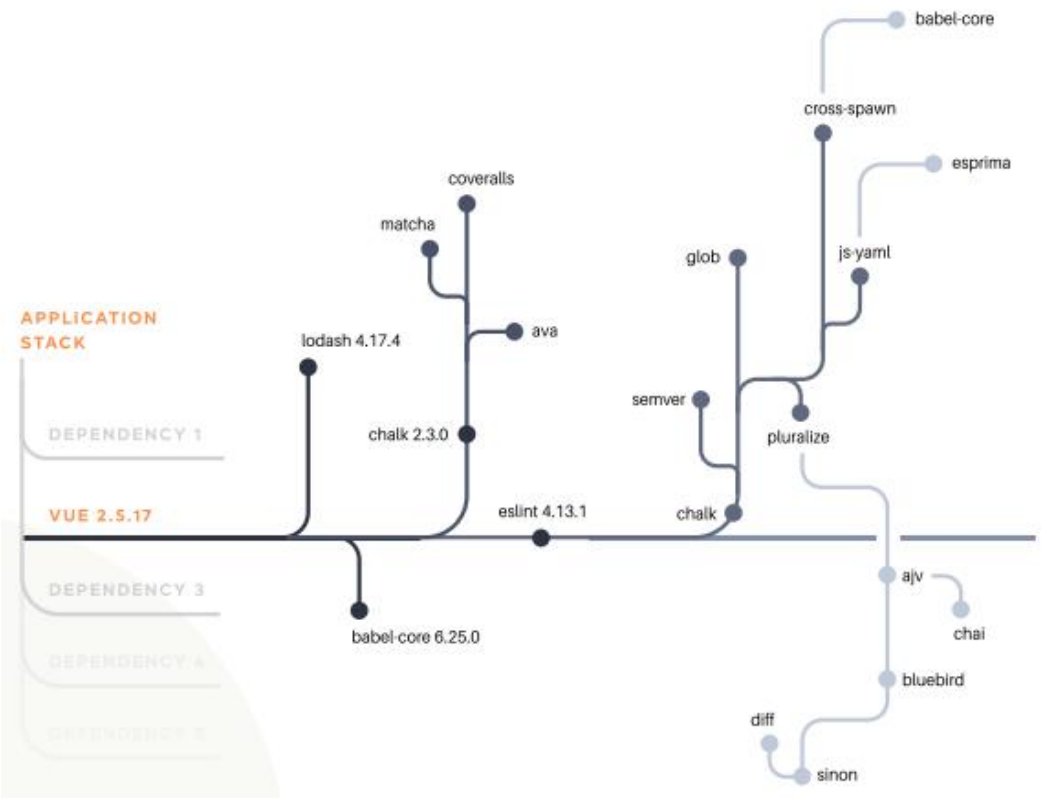
源信息，并且还在计算！）。我们通过索引超过 37 个软件包管理器生态系统以及 GitHub，GitLab 和 Bitbucket 来完成它。把它看作是一个开源软件的搜索引擎。一种几乎可以找到所有可用的开源包并了解它们之间如何相互关联的方法。这使我们想到……

理解软件项目中的依赖关系

软件应用程序总是通过组装许多小软件来构建的，这些小软件通常以软件包的形式交付，或者称为组件或库。当一个组件被添加到您的软件应用程序中时，您的应用程序依赖它来执行一些功能，而这个组件现在是您的应用程序的“依赖项”。每个单独的组件可能有自己的许多依赖项，每个组件可能有许多依赖项，依此类推。整套软件组件构成一个依赖树。特定软件项目或应用程序的完整依赖项集，包括依赖项的依赖项等等，称为其传递依赖项。

有多少依赖?

直接依赖是那些你明确选择依赖的;但是直接依赖反过来又会吸引其他包，称为传递性依赖关系。例如，一个典型的 JavaScript 应用程序可能只有不到 10 个直接依赖项，但是大约有 1000 个传递依赖项。应用程序团队中的任何人通常都没有检查这些传递依赖关系。应用程序依赖关系树的质量可能意味着应用程序是砖房或者是纸牌屋之间的差异。例如，JavaScript 前端包 Vue 的依赖树如下:



极灿科技有什么帮助?

极灿科技跟踪和分析了超过 5 亿个开源软件包依赖关系。使用这些数据，极灿科技将连接到您的软件开发过程了解您使用的确切包。然后我们付钱给这些包的维护者，以确保它们得到专业和积极的维护。

极灿科技有什么独特之处?

我们付钱给维护人员。

剧透警告——今天，维护大多数开源软件项目的报酬并不高，除非在谷歌、Facebook 或其他

支持开源工作的类似公司工作。这对专业开发团队的影响是，大多数维护人员缺乏保持其包维护良好和安全所需的关键激励。
为了开源维护者和专业开发团队的利益，极灿科技的使命是改变这一点，他们将他们创建的软件组件整合到他们的业务应用程序中。
我们认为流行的开源项目的维护者都应该得到报酬，而这些维护者是专业维护项目的最佳人选。
极灿科技付钱给开源维护者来支持他们的项目，提供必要的维护和更新—包括安全性和许可问题——作为极灿科技订阅的一部分。
有了极灿科技，当维护人员消失或不能花时间在依赖项上时，你就可以轻松地替换或更新密钥了，重要的商业担忧已经过去。
开源软件很棒，但它仍然是软件
应用程序使用的每个开源依赖项代表一个独特的软件，具有自己的许可证，发布生命周期以及安全和维护实践。
只要您的应用程序在依赖关系树中的任何一个开放源码包中出现一个主要问题，就会导致安全漏洞，从而有可能暴露客户数据。或者维护问题可能导致客户端面临系统中断。或者，开源项目许可证的问题可能会导致对您的组织提出昂贵且耗时的知识产权侵权索赔。一个简单的事实是，很少有开发团队有时间检查他们的应用程序所依赖的数千个开源项目，以发现这类问题。
但不这样做，也没有一个有效的方法，意味着一些关键的变化可能会在你不知情的情况下发生，例如：
➜领导维护者可能会失去兴趣，获得全职工作，或者出于某种其他原因停止重要依赖
➜依赖关系可能会更改其许可证，或者开始使用具有问题许可证的传递依赖关系
➜可能会出现安全漏洞
➜更新可能会破坏您需要的 API
➜维护者可能会以影响您的方式更改项目路线图
极灿科技有什么帮助?
极灿科技 订阅可以扩展以监控，修复和主动改进您每天使用的众多开源软件包
在下一节中，我们将深入讨论有关安全性、维护和许可证的问题，并演示极灿科技如何提供帮助。
安全性
与所有软件一样，开源软件组件也容易受到影响，由编程疏忽甚至故意破坏造成的安全漏洞。政府赞助的项目，如常见漏洞和漏洞 Exposures（CVE®）编制公共网络安全漏洞的公共标识符列表，包括开源软件中的漏洞。
然而，很少有商业组织拥有健壮和有效的流程来确保他们所有的软件都被持续地评估已知的漏洞。由于开放源码包的数量之多，开发人员几乎不可能跟踪每个包各自可能带来的安全风险。
一个令人痛苦而熟悉的例子是 2017 年 Equifax 的利用。虽然很容易将矛头指向 Equifax，但同样值得一问的是，您自己的组织是否拥有跟踪所有应用程序及其开源依赖项中的漏洞的工具和流程。正如我们已经看到的，单个应用程序中的依赖项数量很容易达到数百个。
“Equifax 证实，攻击者是在 5 月中旬通过一个网络应用程序漏洞进入其系统的，该漏洞在 3 月份就有补丁可用。换句话说，这家信用报告巨头有两个多月的时间来采取预防措施，防止 1.43 亿人的个人数据被曝光。然而它没有。”---《连线》杂志
修补这个安全漏洞需要耗费大量人力，也很困难，部分原因是它需要下载一个更新版的 Struts，然后用它重建所有使用旧的、有 bug 的 Struts 版本的应用程序。一些网站可能依赖

于数十个甚至数百个这样的应用程序，这些应用程序可能分布在多个大陆的数十台服务器上。一旦重新构建，这些应用程序在投入生产前必须经过广泛测试，以确保它们不会破坏网站的关键功能。”--Ars Technica，未能修补两个月的漏洞导致 Equifax 在 2017 年 9 月遭到大规模入侵

极灿科技有什么帮助?

我们与开源维护者合作并补偿他们，以负责任地处理安全问题。具体来说,极灿科技:

➜采取积极的措施,以避免未来漏洞和处理任何新的漏洞

➜告诉您有关影响依赖关系的任何漏洞

➜提供过去的弱点和受影响的版本范围的信息

维护

从本质上讲，软件维护意味着定期添加特性和修复 bug。随着开源包的流行，用户对新特性和修复程序的请求数量也在增长。通常，这种大量的拉请求会超过兼职维护人员的能力。包维护中的失误常常成为试图使用包的专业开发人员的负担。

这些专业开发人员使用的开源包会以不同的频率进行更新和修补，有时需要进行的更改会破坏与应用程序堆栈其他部分的兼容性。这些更新和补丁需要在整个堆栈中进行一系列的更改，以保持一切正常运行。

一点点腐烂和积累的技术债务

当构建，测试和部署它所需的依赖关系和工具随时间发生变化时，所有软件都会发生误操作。最终，当软件需要更改或重新部署时，由于与周围不断变化的生态系统的冲突，它无法返回到正常状态。

软件并不存在于真空中。应用程序构建在数百、甚至数千个开源框架和库的不同部分之上。它们是用各种编程语言编写的，运行在各种操作系统上，并部署到各种硬件上。

所有这些组件都会以不同的频率更新和修补，有时需要进行更改，这会破坏与应用程序堆栈其他部分的兼容性。这些更新和补丁需要在整个堆栈中进行一系列的更改，以保持一切正常运行。

有时，您可以控制何时应用这些更改—例如，您使用的编程语言的版本。有时候你就没那么幸运了，即使是很小的更新也会迫使您在整个应用程序中进行较大的、中断的更改，这只是为了保持功能和安全性。这将影响正在进行积极开发的项目，以及那些多年来在服务器上悄无声息地运行的项目，这些项目似乎没有问题。

什么是一点一滴的腐烂 ？

一点一滴的腐烂是指随着时间的推移，由于构建、测试和部署所需的依赖关系和工具的更改而导致的软件性能的恶化。如果不进行检查，一点一滴的腐烂最终会导致软件经常出现故障，性能低下，或者无法使用。

一点一滴的腐烂往往是千刀万剐的结果;您所依赖的每个软件片段都可能受到任何数量的更改的影响，这些更改可能导致您的应用程序崩溃。

一些最常见的原因是:

➜禁用/更改不安全接口的安全版本

➜错误修复版本，无意中导致 API 更改

➜旧版本已经过生命周期，不再进行兼容性测试

➜主要版本中不一致的重大变化

➜应用程序的依赖关系树中的冲突

➜不可重复的安装步骤阻止您重现工作环境（也称为一次性）

➜第三方或远程 API 在没有事先警告的情况下更改或变为不可用

腐败经常与技术债务的积累密切相关。交付产品的压力可能导致短期的设计和编码妥协。
在《重构:改进现有代码的设计》一书中，作者 Martin Fowler 总结了如下技术债务:
“如果你能在今天完成今天的工作，但你这样做是为了明天不能完成明天的工作，那么你就失败了。”
太多此类的妥协导致了技术债务。您积累的技术债务越多，您的应用程序就越能感受到腐烂的影响。维护问题（如腐烂和积累技术债务）对业务的影响是，软件团队浪费了不可估量的时间来追逐和解决软件组件兼容性问题（并且他们可能根本无法解决这些问题），而不是解决他们的实际业务问题。

由此产生的业务影响是安静但痛苦的现实：

➜降低开发人员的工作效率
➜昂贵的软件团队工资浪费在琐碎但耗时的问题上
➜开发者产生挫败感和留住员工的挑战
➜需要更长时间才能完成并超出预算的软件项目
极灿科技有什么帮助?
我们与开源维护者合作并补偿他们专业地维护他们的软件，并对其未来的可靠性做出承诺。
具体来说,极灿科技：
➜与开源维护人员合作，确保他们理解您的问题，并且帮助您解决问题
➜根据一系列统一的最佳实践补偿维护人员积极维护他们的包裹 - 查看即将发生的问题，考虑如何进一步开发包装，并保持开发正常进行
➜分析您的依赖关系以确保标记维护问题，以便您可以在必要时进行更改
➜通过“依赖项中的新功能”活动源为您使用的项目提供发行说明
➜帮助您了解单个软件包装稳定性保证和分支方案
谁是开源维护者？--遇见更多像埃文这样的

-埃文没有为他流行的 JavaScript 框架做计划，Vue.js 成为一份全职工作。 就这样发生了。
-埃文是在研究生毕业后在谷歌创意实验室工作时想到 Vue 的。
“我们在一些项目中使用了 Angular 1，“Evan 说。Angular 1 中有我喜欢的部分，也有我不需要的部分。Vue 最初是一个提取部分内容的实验——专门用于构建 web 应用程序。”Evan 说“我们正在尝试从更整体的完整堆栈框架中获取这些优势，但我们希望为那些为更简单的用例构建的人提供这些内容。”

从 2013 年底到 2016 年初，Vue 是一个为期两年半的项目.Evan 不得不休假三周，投入全部时间工作，发布 1.0。 大约在那个时候，当创作者 Taylor Otwell 开始使用时，Vue 被 Laravel 社区所接受。
“(当 Otwell)开始使用 Vue 时，整个 Laravel 社区都支持他，”Evan 说。“我们看到使用量激增。就在那时，我想，‘也许这里面有比副业更大的东西。’
当他的日常工作开始朝他不感兴趣的方向发展时，Evan 决定辞职，全职做 Vue。
对于像 Evan 这样的维护者，极灿科技提供了稳定的收入来源，因此他们可以使他们的开源项目更加出色。

许可证

软件许可和知识产权法是使开源软件得以存在的关键“技术”。
每一个开放源码软件都保证您能够使用它，但是至少对如何使用它有一些最小的限制，可能还有属性要求。
大约 10%的软件包有更严格的限制，通常称为“copyleft”。
然而，特别是在具有许多依赖关系的大型软件项目中，开源许可证合规性可能成为一个主要的复杂事实。

什么是 COPYLEFT 许可证?
“Copyleft”或“互惠”许可证要求在特定条件下共享修改。即使对于拥有大量资源的公司来说，开放源码和许可证也可能是一个非常头痛的问题，因为许多不同的许可证可能彼此不兼容。对于任何事情，仅仅想要获得准确和完整的许可信息都是一件费时费力的事情，尤其是在大规模的情况下。与此同时，公司还需要更好地了解遵守开源许可证的问题
----新的栈，SPDX 可以帮助组织更好地管理他们的开放源码许可证

通常，单个软件开发人员的任务是对其开源组件的软件许可证做出关键决策——这是一个复杂而微妙的专业领域，他们没有接受过相关培训。
由此产生的业务影响可能很痛苦，包括：
➜第三方因违反许可或版权而遭受诉讼
➜在软件开发过程的后期必须删除组件以修复许可问题时的昂贵和非计划成本
➜与更改已部署的产品相关的客户服务和支持成本增加，开发速度降低
➜延迟或直接取消融资,兼并和收购,首次公开发行(ipo)和其他公司交易
➜法律禁止继续发布有未决侵权索赔的产品

对于不擅长处理开放源码的公司来说，合规性也是一个令人头痛的问题。确保您的公司分发源代码并完全遵守 GPL，包括确保如果 GPL 与其他兼容许可证的作品相结合，则需要相当多的关注 -- CIO 杂志，开源许可如何影响您的业务和开发人员

极灿科技有什么帮助?
我们与开源维护人员合作并对其进行补偿，以帮助用户避免知识产权和许可问题。 具体来说，极灿科技：
➜确保属于极灿科技 订阅维护合约的所有软件包均在经批准的开源许可下运行
➜清除许可证元数据，确保每个包的一致性和准确性
➜修复许可证违规

➜帮助您了解您是否具有可能导致典型专业开发用例问题的限制性许可的依赖性
➜获得主要维护者的保证，他们没有侵犯他人的权利，并且您将有一个窗口可以解决无意中的错误
第二步：事故预防
评估您的组织当前对开源的使用情况
一旦您理解了在专业开发环境中使用开源所涉及的问题，下一步就是通过映射依赖关系树来全面了解组织的开源使用情况。
虽然这曾经是近乎艰巨的任务，但现在由于极灿科技而变得更加容易。 下面是如何开始：
获得开源评估。
通过支持 Javascript，Java，Python，PHP 以及 20 多种语言和包管理器，我们的开源软件评估将为您提供组织已使用的所有开源组件的统一视图。
极灿科技开源评估揭示了关于您的堆栈的重要信息，包括:
➜已弃用和未维护的包
➜缺少许可证
➜安全漏洞
➜直接和传递依赖树
制定计划
现在您已经拥有了组织使用的开源组件的完整列表，并且可以更清楚地了解当前的任何问题，现在是时候制定解决这些问题的计划了。.
一条前进的道路是通过升级单个软件包，提供自己的 pull 请求 toupstream 开源软件包，或者从应用程序中删除有问题的依赖项并选择其他软件包，或者自己从头开始实现类似功能来开始尝试修复问题。 然后，盯着仪表盘，冲洗，重复。
但在极灿科技，我们提供了一个更全面和可扩展的模型。当您购买极灿科技订阅时，您可以依赖与极灿科技合作的专家维护人员来为您完成这项工作，以专业的方式修复当前可见的问题，并主动避免和解决将来出现的问题。
第三步：决议
使用极灿科技实施专业的开源管理计划
大多数组织都有软件供应商管理计划，以保护他们免受风险，并确保他们的软件得到适当维护。供应商管理程序通常由采购或法律强制执行，确保组织内使用的软件是可靠的。供应商管理程序要求软件供应商提供有关未来发现或可能出现的问题的陈述，并采取正式手段来纠正这些问题。
但谁为开源软件提供代表？
数以百万计的高质量开源组件的出现超过了组织跟踪其使用该软件的能力。所有软件、开源组件都需要维护，理想情况下还应该包含关于软件安全性、可靠性和知识产权所有权的其他表述。
从历史上看，通常没有人为专业开发团队使用 dai 的绝大多数开源组件提供这些保证。
这是一个错失的机会，使专业开发团队陷入长期维护和维护困境，并且容易受到许多不良后果的影响。有超过 330 万个开源软件包，任何组织想要做的最后一件事就是沸腾海洋。 相反，专业开发团队应该只关注他们的组织所依赖的开源。
极灿科技有什么帮助?
极灿科技订阅使组织可以采用专业的方法来轻松实现开源软件的管理，使其更安全，更可靠。
➜我们汇集的维护者,您已经在使用的组件,并继续适应您添加和删除组件
➜我们与维护人员合作，建立专业级维护，安全和许可标准和流程，作为极灿科技订阅的一部分

➜通过极灿科技订阅，我们为您的组织提供与任何其他软件供应商相同的可靠性表示。
➜您的团队与一个供应商合作，节省时间和金钱。您的开源依赖关系得到了广泛的覆盖。