链滴

史上最全的开源项目创作指南

作者：xuexiangjys

原文链接：https://ld246.com/article/1606235751576

来源网站：链滴

<h2 id="前言">前言</h2>
<blockquote>
开源，是这个时代的主旋律。作为一名 Android 开发工程师，我有理由相信我们是开源的最大益者，因为那个养活我们的 Android 其本身就是 Google 的开源项目。在这样一个开源的时代，就之前那个最排斥开源的"微软"也不得不积极拥抱开源，大手笔收购 github 以表自己的开源决心。</p

</blockquote>
前段时间我在逛 github 的时候，偶然间发现，我的 github 已经拥有 12 个 star 过百的开源项，2 个 star 过千的项目。回首一想，原来我做开源项目已经快 3 年了，想想这一路走下来真的非常易。

深夜和周末是我做开源项目的主战场，电脑和 AndroidStudio 是我创作的纸笔，各种 bug 和 isses 是我通往卓越项目的拦路虎，经常为了解决一个问题独自思考到凌晨。这其中还要饱受喷子们的挖和质疑，以及白嫖党的夺命连环 call。当然，也会有人站出来说几句公道话，但是平心而论，国内的源环境真的很差。
如果不是因为对技术的热爱以及支持者的鼓励，我相信我很难坚持下去，就像那些渐渐消失的开项目创作者一样，提交记录永远地停留在了那一刻。
所以，为了能够记录下我的这段艰苦开源之旅，同时也是希望能够改善国内的开源环境，帮助更希望从事开源项目的有志青年，我决定写下这篇开源项目经验总结。
<h2 id="为什么做开源项目">为什么做开源项目</h2>
<blockquote>
在决定做开源项目之前，你非常有必要问一下自己：我到底为什么要做开源项目？无论是出于什目的，只要你有答案的话，那么你就可以继续往下看，否则以下的内容可能对你来说没有任何意义。
</blockquote>
说起开源，就不得不提 Google 这家极具开源精神的公司。作为一家美国科技公司，每年都在不地对外输出着无数优质的开源项目，与此同时，近几年我们国家的科技公司们也开始着手开源计划，源了不少有趣的项目。
那么为什么大型的科技公司都在积极地做开源项目呢？其实原因很简单，无非就是为了名气和企的形象嘛。作为一家市值超百亿的科技公司，不搞点开源项目出来装装逼，都不好意思说自己是大厂
那么作为我们个人开发者而言，有必要做开源项目吗？在回答这个问题前，你有必要问一下自己你真的热爱做技术吗？如果说你做技术的目的只是为了养家糊口的话，我觉得接接私活，做做外包比适合你。因为做开源项目真的是那些"闲得蛋疼"的人打发时间的玩具，在这条道路上你会发现非常没有钱途"。只有当你真正将技术作为一种兴趣来热爱的时候，你才能体会到那种开源项目被无数人引用的价值和喜悦感。
所以，说了这么多，我们究竟为什么要做开源项目呢？以下列举我的几点理由，供大家参考：</>

<code>提高自己的技术水平。</code> 毕竟开源项目就相当于把自己扒光了给别人看，也许样比喻不是很恰当，但确实能提高自己代码的质量和解决问题的能力。毕竟如果你的开源项目真的有用的话，你必然会收到很多的 issue 以及建议，这些东西可能你之前根本就没有想到。
<code>让项目变得更加健壮。</code> 开源的最大妙处就在于，任何人都有权利看到你项目代码，提出自己的建议。这其中就有可能发现项目中存在的漏洞，以及一些非常有建设性的建议，同也能让你明白自己项目中存在的不足，这样就会推动项目不断的优化升级，项目的质量也会呈螺旋型升而变得更加健壮。
<code>结交志同道合的朋友。</code> 你在做开源项目的同时，很容易就能结交到与自己志道合的开源创作者，毕竟大家都是希望开源项目越做越好，很有可能你的开源项目被其他的开源者所识而一同开发和维护。这样一来一去你们很有可能成为"同志"，哈哈。
<code>帮助他人，提升自己的行业影响力。</code> 这也是很多人做开源的重要目的。如果有幸能够在某个领域做出一个非常优质的开源项目并且有很多人引用的话，那么你在该领域的名气一会蹭蹭的往上冒，到那时什么都会有的，具体可参考 vue.js 的创作者：尤雨溪。

<code>展示个人的技术水平。</code> 很多人做开源项的目的就是它能够在应聘面试中加分尤其是应聘某些大厂。因为通过那短短几小时的面试其实并不能很全面地考察出一个人的技术水平，且面试通常仅仅是两个人面对面空谈几小时，如果这个时候你能拿出一个较为优秀的开源项目的话，定会加分不少，至少能在面试前给面试官营造一个不错的印象。
<code>实现自己的人生价值。</code> 开源给予了你无穷大的创作自由，在上面没有产品经提出的各种脑残需求，也没有测试提出的各种过度测试 bug，更没有各种技术条条框框以及企业规范你可以做任何你认为对或者感兴趣的事情；你可以实现自己儿时改变世界的梦想；你可以留下你在这世上存在过的印记；你可以...开源给予你无穷大的舞台，你可以做很多有意义的事情，当热前提还是能违法法律...触犯法律的事情我们不能做，毕竟还是保命要紧，哈哈。
<code>收获意想不到的业余收入。</code> 虽然开源是免费的，不应以盈利为目的。但是不不说没有经济效益的驱使，单纯靠对技术的热爱其实是很难坚持下去的，尤其是当你面对很多现实生的压力时，情怀就显得那么苍白无力，这也是为什么各大开源平台都为开源者提供了赞助和打赏的渠，毕竟开源者也要赚钱生活。好的开源项目其实是能够收获相当可观的打赏的，更有甚者可以搞一个用版，授权使用，赚取服务费等，当然能做到这一步的开源项目还是不多的。

<h2 id="如何做好开源项目">如何做好开源项目</h2>
<blockquote>
上面我简单讲述了我们为什么要做开源项目，如果此刻你心中有了答案，那么恭喜你，你已经成一名准开源创作者，那么下一步我们就是探讨如何做好一个开源项目。
</blockquote>
我做了近 3 年的开源项目，其中最为成功的 XU 近 600 次的代码提交，齐全的文档和视频教程，目前也不过收获了 2.7k 的 star 量，所以说做一个开源项目其实是非常不易的。
下面我就简单拿 XUI 项目为例，简单介一下衡量一个开源项目质量的几项指标。
<img src="https://ld246.com/images/img-loading.svg" alt="open_project.png" data-src "https://b3logfile.com/file/2020/11/7b729234f62e4f2894985a558e826664.png?imageView2/2interlace/1/format/jpg">

项目的热门程度：项目的 star、fork 和 watch 量。
项目的活跃度：这里考量的因素包含 issue 的总体数量、open issue 和 closed issue 的数量、issue 回复和解决的速度、pull requests 的数量、项目最后一次提交的时间。
文档是否齐全：是否有 wiki 或者 README.md。
项目的稳定性：代码提交的频率，项目版本发布的频率。
项目的潜力：项目开发的分支数、项目的开发计划以及项目参与者的数量等。
项目代码的质量：设计是否合理，是否符合设计模式原则，考虑项目的可扩展性、便利性和稳定。
开源作者的水平：作者其他项目的 star 量和行业影响力。

只有了解了以上指标，我们才能创作出更加优质的开源项目。那么说了这么多，我们如何才能做一个开源项目呢？请继续往下看！
<h3 id="1-选对开源托管平台">1.选对开源托管平台</h3>
开源托管平台以 github 作为首选，云 作为备份。
虽然目前市面上开源项目的托管平台非常多，比如：码云、码市、gitlab、BitBucket、SourceFoge，不过我还是极力推荐 github，毕竟 github 使用的人群最广，人数最多，谁不想自己幸苦创作开源项目能被更多人看见呢。虽然 github 是一家美国企业，日后有被禁掉的风险，但是我相信一个尚自由、民主的国家，对禁开源平台这件事情还是不会那么顺利的，毕竟开源无国界，开源不应政治，商业化！如果你还是担心 github 日后会被禁，那么很简单，你直接把你的 github 项目一键导入码云中作为备份，毕竟码云是得到国家认可的，还是比较可靠的。

<h3 id="2-好的创意或者理念">2.好的创意或者理念</h3>
如果一项功能、一件事情大家每次都需要重复去做，但是又没有什么好的解决方案或者轮子的话这个时候我们就可以尝试去做一个。
一个好的开源项目都是为了解一个问题而诞生的。如果你有好的创意或者理念，那么你就更能吸更多的人参与到项目的建设中，那样也会有更多的人关注到你的项目，这样你的项目想不火起来都难
我当初创作 XUI 就是希望能够简化 Androi 界面开发的难度，提升 Android 界面开发的效率而做的尝试。相信做过 Android 的人都知道 Andro d 原生组件在国内很不受设计师的待见，至于 Google 推行的 Material Design 设计风格更是无人问，这就导致了设计师给出的原型图几乎是清一色的 IOS 风格，更尴尬的是，网上 Android 相关的开源 UI 库是少之又少，几乎所有的基础组件都需要自己重写。正巧当时我接触到了 React 和 Vue，发现们都有非常方便的 UI 库，直接在示例代码的基础上修修改改就能大致上实现自己想要的效果，极大提高了开发的效率,后来我又借鉴了 QMUI<a> 相关的思想，最终创作出了 XUI 这个开源目。
所以，一个好的创意或者理念对于开源项目来说非常重要，可以说是开源项目的灵魂。
<h3 id="3-好的设计和代码质量">3.好的设计和代码质量</h3>
<blockquote>
如果说一个好的创意或者理念是开源项目的灵魂的话，那么一个好的设计和代码质量就是开源项的骨骼和肉体了。
</blockquote>
<h4 id="3-1-掌握设计模式">3.1 掌握设计模式</h4>
要想有好的设计，首先你需要非常熟练地掌握 <code>设计模式</code>，那么如何才能熟练掌握 <code>设计模式</code> 呢？在这里我可以教大家一些经验：

1.首先了解设计模式的几大基本原则。这里我有一篇讲关于设计模式原则的博客 可供大家参考。
2.其次初步学习现有的二十几种设计模式，并在平时的工作或者开源项目中尝试使用。这里我有个专门介绍设计模式使用的开源项目 ,里面有相应的介绍和源码可供大家参考。
3.最后等你熟练使用了上面的二十几种设计模式后，忘掉他们，重新回顾之前的设计模式基本原，并以此作为日后项目设计的基本原则。

其实学习 <code>设计模式</code> 非常像武侠小说中修练一门武功，学习设计原则是修炼心、内功，而学习现成设计模式则是修炼招式。只有提升内功，牢记心法，忘记招式才能真正意义上掌了 <code>设计模式</code> 这一项技艺。
<h4 id="3-2-严格的代码规范">3.2 严格的代码规范</h4>
提高代码质量最简单的途径就是严格遵循通用的代码规范，这里我推荐阿里巴巴 Java 开发手册 和它的 IDE 插件 <a href"https://link.ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Falibaba%2Fp3c" targt="_blank" rel="nofollow ugc">p3c 项目 。
只有遵循通用的代码规范，这样才更加利于开源项目的多人协作，除非你想一个人维护整个项目否则你的代码写得那么骚，谁能看得懂？
<h3 id="4-丰富的案例或者测试用例">4.丰富的案例或者测试用例</h3>
作为一个合格的开源项目，提供一些单元测试用例还是非常有必要的，因为你写出来的东西并没专门的人给你测试，这个时候如果还没有相应的单元测试用例，你如何保证你写出来的东西不是个坑

？
如果你的项目不太适合写单元测试用例的话，那么你最好能提供丰富的使用案例，这样才能让你开源项目更具吸引力，让别人有东西可以上手实践，否则光秃秃的啥也没有，你写这个项目的意义又哪里？
<h3 id="5-完善的文档">5.完善的文档</h3>
这里的文档主要包括 <code>README</code>（简介）和 <code>wiki</code>（使用文档。下面是文档的几项基本要求：

1.文档内容应当尽可能的简洁明了，层次分明，并且将其放置于显眼的位置，便于他人寻找。
2.文档应当及时更新，避免出现文档与代码不一致的情况。
3.提供多语言版本的文档，至少英文版文档是要有的。

<h4 id="5-1-README编写">5.1 README 编写</h4>
<blockquote>
README 可谓是开源项目的门户，每个人都是从阅读你写的 README 开始了解你的开源项目。README 写得好与坏，可能直接关系到别人在你的开源项目主页上停留的时间，以及他们是否会你的项目一个小星星，记住这里很关键！
</blockquote>
如果你之前从来都没有写过 README，那么这里我推荐一个国外人写的教科书式的项目 Standard Readme ,里面写的内容非常经，可以拿来借鉴。
那么，一个写得好的 README 应当包含哪些内容呢？以下是我总结出来的经验仅供参考：

项目简介（必须）：用几句话简要描述你的项目特点、优势以及项目目标（解决何种问题）。
作者简介（选填）：这里是你推广自己或者其他项目的地方。
项目背景（选填）：这里可以写你创作该项目的缘由和过程。
项目特点（必须）：这里是你区别于其他项目的关键，可以写你项目独有的功能、优势、理念等
项目设计（选填）：这里可以阐述项目的设计思想和理念。你可以借助流程图、思维导图、UML类图、时序图等方式进行阐述。
项目演示（必填）：这里是别人能最直观感受项目魅力的地方。你可以通过以下 5 种途径进行演。

gif 动画演示
视频演示
图片演示
在线演示
Demo 下载

集成/安装指南（必填）：这里你得告诉别人如何才能快速地使用上你的项目。
使用文档（必填）：这里你需要告诉别人该如何正确使用你的项目，越详细越好。当然如果内容多的话，还是建议直接给一个使用文档首页的链接或者视频教程，这样可能会更友好一点。
相关项目（选填）：这里是你推广其他开源项目的地方，是让你的项目粉丝数产生裂变的地方，定不要错过。
如何贡献（选填）：这里的贡献包括两个方面：代码贡献和打赏和赞助。对于代码贡献，你需要出你的提交规范和行为准则，以避免不必要的麻烦。
特别感谢（选填）：这里写上对你项目有过帮助或者启发的人或者项目的地址，以此表达对他们感谢。
联系方式（选填）：这里可以写你为此项目创建的 QQ 技术交流群、微信公众号等。
许可声明（选填）：这里你可以写你项目的开源协议或者许可使用的范围（比如：禁止用于商业

途，仅供学习）。如果你分免费版或者商用版的话，也可以在这里进行说明。

具体内容你可以参考 XUI 项目的 README 或者我的 README 模版 。</>
<h4 id="5-2-wiki编写">5.2 wiki 编写</h4>
<blockquote>
wiki 的编写和及时更新非常重要。wiki 最好分模块进行编写，做到条理清晰，层次分明，通俗懂。
</blockquote>
下面我就以我的另一个开源项目 Xpdate 的 wiki 来简单说一下，wiki 我们该如何编写。如果你会在线文档编写的话，这里可以直跳过。
<img src="https://ld246.com/images/img-loading.svg" alt="wiki_example.png" data-src"https://b3logfile.com/file/2020/11/c5dfa1243e8f412192fc31ede573afcf.png?imageView2/2/iterlace/1/format/jpg">
wiki 主要可以分为三块，如上图所示，最上面是项目的简要描述，左侧是文档的首页，右侧是文的目录。这里我为了偷懒，左侧的文档首页直接照抄了右侧的文档目录。
下面我简单列举我们的 wiki 中应当包含哪些内容：

简介：虽然这里的简介绝大多数是和 README 重复的，但是最好还是不能漏。

项目简介：这里包括项目描述、项目特点、项目背景、项目设计思想等内容。
集成/安装指南：如果你的项目是轮子库，那么就编写集成指南。如果你的项目是完整项目，那就编写安装指南。
项目演示：gif 动画、视频、图片演示，在线演示和 demo 下载二维码任选几项提供即可。

如果使用

基础使用：顾名思义，能够满足最基础需求的使用介绍。
进阶使用：这里主要介绍一些个性化（自定义）、深入使用的操作。
常见问题：常见问题的整理非常重要，它往往产生于 issue 或者技术交流群中常见的问题，对刚触该项目的人非常有用。

配套设施：一个优秀的开源项目往往会自建相应的生态，这时配套设施就非常有必要了。

<h3 id="6-合理的版本管理和规划">6.合理的版本管理和规划</h3>
<blockquote>
新版本的发行不宜过于频繁，也不宜间隔时间过长，胡乱无规律的版本发行对使用者而言简直是灾难。
</blockquote>
最恰当的做法应该是：在保证充分测试没有问题之后，定期发布最新版本，实时更新项目的最新度以及未来的开发计划。
这里我提供几点建议供大家参考：

每次在进行新版本的开发时，建议单独拉一个 dev 版本分支，减少对主分支的影响。<img src="ttps://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2020/1/abf99b73b6064eefa841f5de423eaee6.png?imageView2/2/interlace/1/format/jpg">

可以使用 github 上的 projects 来规划你的项目开发计划。<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2020/11/46f3c5023c9c4aa9b3d0e8ed6830115.png?imageView2/2/interlace/1/format/jpg">
可以在 issues 中建一个置顶的 issue 来收集使用者的反馈，以此作为新版本开发的规划素材。<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2020/11/679b1f7afdb3435383a54bb2bc01adc7.png?imageView2/2/interlace/1/format/jpg">
每次新版本发布的时候，一定要提供详细的版本日志，方便使用者参考和追踪。同时，对于几个动比较大的版本调整一定要有明确的说明，否则使用者会非常疑惑。<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile.com/file/2020/11/61af47c2f47a4e8926100bbe35a6369.png?imageView2/2/interlace/1/format/jpg">

<h3 id="7-关注反馈-持续更新">7.关注反馈，持续更新</h3>
<h4 id="7-1-及时处理issue">7.1 及时处理 issue</h4>
<blockquote>
issues 是使用者与项目开发者之间沟通的桥梁。很多使用者提出的 issue 还是非常有建设意义的及时高效地处理掉它们，可以让我们的项目变得更加完美。
</blockquote>
在处理 issues 的过程中，我们可以收集整理"常见问题"，收获好的 idea，了解自己项目存在的足等。这就要求我们需要及时关注和处理使用者提出的这些 issues。
那么我们怎样才能快速高效地处理 issues 呢？这里我提供几点建议：

1.提供 issue 模版，过滤无效 issue，提升沟通的效率。这里你可以参考 XUI 的 issue 模版 。

<img src="https://ld246.com/images/img-loading.svg" alt="issues_template.png" data-src="https://b3logfile.com/file/2020/11/48e4cf80218a4be2bfa21f5798264ffe.png?imageView2//interlace/1/format/jpg">

2.自定义 issue 标签，对 issue 做分类管理，部分 issue 可以优先处理。

<img src="https://ld246.com/images/img-loading.svg" alt="issue_labels.png" data-src="ttps://b3logfile.com/file/2020/11/1839bb7a74734bc584bd63df5b9cea82.png?imageView2/2/nterlace/1/format/jpg">
<h4 id="7-2-欢迎PR-及时处理PR">7.2 欢迎 PR，及时处理 PR</h4>
在开源项目初期，靠一个人维护一个项目还是不难的。一旦项目热度上去来，光靠一个人的精力远远不足以维护整个项目的，这里我深有体会。
所以我们需要提供 PR 提交规范和行为准则，积极欢迎更多的人参与到项目的维护中。同时对于人提出的 PR，我们要及时 review 验证，对于没有问题的提交应当适时合入以提高别人参与贡献的积性。
<h3 id="8-做好推广">8.做好推广</h3>
<blockquote>
酒香不怕巷子深，这句名言在互联网时代是行不通的。你就是有再好的创意、再好的设计，如果做好推广的话，也是没人能看到你的开源项目的。
</blockquote>
如何推广自己的开源项目，对此我还是非常有经验的，下面我就提供几个途径：

在各大技术论坛、博客、社区不断地发文章介绍自己的开源项目。我常用的几个平台有：CSDN 、掘金 、<a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fsegmentfault.com%2F" target="_blank" rel="nofollow

gc">思否 、简书 、知乎 、哔哩哔哩 。
在一些开源项目推荐分享平台上提交自己的项目请求收录。比如我的开源项目 XUI 和 XUpdate 提交到了 HelloGitHub 上。
组建自己的开源社区。你可以创建 qq 交流群、微信公众号或者社区网址来推广你的项目。
加入知名的组织。比如我就加入了 B3log 开源社区 。

<h3 id="9-不忘初心-摆正心态">9.不忘初心，摆正心态</h3>
做开源项目是一件非常漫长的过程，你可能根本想象不到前方的道路有多么的曲折。
假如你的开源项目做了几个月了也无人问津，不要气馁，专心去做你认为有价值的事情即可，总一天会有人发现你项目的价值。
假如你的开源项目做成功了，在收获到不少人赞赏的同时，那么势必会遭到很多喷子们的冷嘲热以及各种无脑喷。不要理会那些只会喷但是啥也不会做的麻瓜，不要把你有限的精力放在这些人身上请把你有限的精力放在那些给你开源项目提出宝贵建议的人身上，专心去做你认为对的事情。
<h2 id="最后">最后</h2>
我花了整整一周的时间整理才写下了这篇文章，也是真诚地希望能够改善国内的开源环境，帮助多希望从事开源项目的有志青年。如果你觉得有用的话，建议你收藏此文章。最后，还是祝愿大家能日写出属于自己的优秀的开源项目！！！
<h2 id="微信公众号">微信公众号</h2>
<blockquote>
更多资讯内容，欢迎扫描关注我的个人微信公众号:【我的 Android 开源之旅】

</blockquote>