

# 《开源软件设计与开发》课程总结

姓名：赵威

学号：52195100008

## 1、开源理解

对于开源的理解，我大概可以分为这样几个阶段，我想现在也许仍有一些人是这样误解的：开源是免费软件，随便使用，没有版权。开源是开放源代码，随便拿来用，想修改就修改。开源是个人开发的，没有公司，也没有版权，并且免费。开源是一群无偿开发的项目，域名大部分是.org 的。开源是谁都可以参加的项目，它由一个委员会主导。

然而现在，我觉得开源是一种去中心化的创造，让我们可以自由的发挥来满足创造欲，我理想的开源项目应该没有 leader 没有核心的，甚至开源可以不是一个项目，它可以是一团代码或者一个半成品，就像自助餐桌上的菜品一样让人随时搭配享用或者二次烹饪，有的只是我们需要“吃”的这种需求。

而说到开源精神，那也许是一种给予和获取的平衡，有给予才能有获取，有获取才会有给予的动力。无需指责别人只会获取，我们应该懂得开源是一种创造方式，一个没有创造欲和创造力的人加入开源也是无用的。可以这么说，如果没有开源社区，苹果，谷歌，Facebook 都不会是现在这样。如果没有 Linux，现在互联网的格局简直无法想象。为了让更多人能够使用优秀的产品，需要有人来愿意把自己的产品免费给他人使用。很多开源社区的贡献者都是为了建立，维持这样一个互助互利的社区而不断努力的。开源一个很美妙的地方就是，别人不用花时间解决一样的问题。当你看到你的一个大问题已经有人做了开源项目的时候，你一定感觉如释重负。同样，有人感谢你的开源项目帮他们解决了一个大问题，真的是种很美妙的成就感。开源社区中经常能遇到很多志同道合的好朋友。你帮我改代码，我帮你加功能，说不定两个人就认识成了好朋友。我和我的一位导师和几位朋友就是在贡献开源项目的时候认识的。

从自私的角度看提升代码质量：开源是提升个人代码质量的好方法。这个有两个方面：首先，如果你知道你的代码要给别人看，你会不自觉的把代码写的更干净一点，多加点说明。其次，开源之后别人会建议怎么改（如果你的项目有一定影响力），给你建议的人多了，你也大概就有概念怎么把代码写的更好了。展示个人代码：如果你在找工作，这个很可能是你的主要动机。公司实际上很好奇你写的代码质量如何，但是光从 1 个小时的面试实在看不出太多。所以很多公司会非常喜欢看你的开源项目，来评估一下你的代码质量如何。同样，如果你面试的时候告诉公司，“我有个开源项目，你们可以去看看”，这将是是个不错的加分项。建立个人品牌：如果你足够厉害，把一个优质项目开源是个迅速提升影响力的好方法。很多在大公司工作过的顶级工程师，我们都是通过他们的开源项目认识的。git 的发明人，Linus Torvalds，就是一个好例子。一个工程师如果在工作之外想要树立个人品牌的话，写一个高质量的开源项目是个相当好的选择。其实很多人把项目放到 Github 的时候根本没想这么多。。。他们只是想把家庭作业，代码练习啥的备份到网上去而已。。。其实 Github 当中绝大部分项目都是这种实际没啥用的代码备份（只是 Github 没让你看到）。公司与个人开源的动机不同。公司开源很多情况下是为了提升外界对公司的形象认知。任何一个大科技公司，肯定用了多多少少用了开源项目（Linux，至少吧？）。如果连一个开源项目都没有，这很容易给人造成，“这公司总是索取，却不回馈给开源社区”的形象（尤其在美国）。比如，我读了谷歌的 Android 的代码，发现里面有个大安全漏洞，然后我提交一个修改把漏洞给补了。你说谷歌想不想拉起我去工作？很多公司一个很好的招聘渠道是开源项目的贡献者。如果能在公

司开源项目做出有意义贡献的话，这个对于公司来说是比较面试好很多的能力证明。这点实际上很有意思，有的公司 / 项目实际上有很多想做的小功能，但是没时间也没人力去单独做。这个时候开源的力量就体现出来了，项目开源之后很多时候能吸引到开源贡献者来做一些自己感兴趣，项目想要，但项目没时间做的 feature。公司开源项目的质量，很多时候能反映公司的影响力。所以硅谷的很多大公司，都有一些厉害的开源项目。一方面是回馈社区，一方面是展示公司的影响力和代码质量，吸引更多人才。同样，如果一个公司很希望自己的一个产品有更大影响力的话，开源是个好方法。苹果的 Swift 就是个好例子。Swift 开源之后，能够用在不少 iOS 之外的领域。说不定之后我们还能看到用 Swift 写的网站架构。

## 2、开源贡献

很抱歉，经过不懈努力，目前还没有对开源做出贡献

## 3、课程反馈

课程整体感觉不错，讲课风格很舒服，但是讲课内容有关技术的部分很少，希望能够在讨论课形式的基础上，传授更多专业性的知识。

## 4、参考文献

无