

《开源软件设计与开发》课程总结

姓名：苏斌 学号：51194507011

1. 开源理解

我们都喜欢开源，在国内一提到开源也都认为是好事情。国内多数软件公司或开发者对于开源的热情，仅仅局限以下两方面：免费的代码；深入参与的开源活动。对开源，我们用的多，贡献少。自由软件精神领袖 Richard Matthew Stallman 的一次演讲中他特别强调“Free is not free”，对这句话我的个人解释是“自由，而不是免费”。开源是自由，但不等于免费。开源是一种商业行为。试想一下，我们平日用谷歌百度做搜索，表面是免费，背后隐藏的是商业广告给他们带来的巨大利润。我们在尽情享用那些开源的免费大餐时，已然完全忘记开源的真正力量——开源是自由的，但不是免费的。

除此之外，对开源定义的理解，也可以借用先鲁迅先生所说的“拿来主义”。把资本主义的资源拿来，给我们社会主义用。而拿来主义并非是不好的，原因是只有当你去用了这个产品，你才能去了解开源，了解他开源的产品到底是什么样的。

真正贡献一个开源产品，需要做到以下几点：理解一个开源产品的应用场景是什么；当初这个作者为什么会设计，设计这个产品当初是为了解决哪一些问题；作者解决的这些场景问题，是否符合我的场景问题；基于上一点的判断，如果处在同样的场景同样的问题才可以使用“拿来主义”。应用也不是浅尝辄止，要深入研究。至少做到把他们的文档读一遍。使用之后发现一些功能不能满足自己的需求，此时可以说，这个开源有缺失，再贡献进去自己的代码，把这个功能完善。

在同个项目中会存在一些功能相似的功能，或者提供了不少的选择给用户来选择。我个人觉得这种做法虽然提高了软件的可扩展性，但带来了接口维护的碎片化，以及用户可能没有办法在一堆选项中找到最佳实践。虽然说不少开源软件是自由的，但在这方面设计来说，我觉得与其提供多个类似的功能，以及多种的选择，其实不如提供一个最佳实践来的好，虽然这样显得没有那么自由，有可选性。但一方面减少了开发维护的难度，另一方面也减少了用户的学习成本。同时，我认为减少一些不必要的功能，也能带来更好的体验。例如博客文章的分类功能，由于这个分类很难成一个标准。用户在写的时候需要考虑要放进什么分类，而读者在阅读的过程很大情况下都是从搜索引擎进来的，他们很可能直接进入某一篇文章，或者在首页从头看到尾。所以相对来说分类不是很重要，反而可能给使用带来麻烦。

以上是我认为的开源的使用方式，很多开源项目也都是这样发展起来的。

截至今年 8 月，开源社区 GitHub 上已经汇集了 4000 多万的开发者、210 万家公司和机构，拥有超过 1 亿的代码库。在 GitHub 上的热门开源项目中，人工智能类占了很大比例。开源开放是人工智能领域发展的全球主要趋势之一。人工智能的快速发展，推动开源开放生态越发繁荣；而开源开放同时也推动人工智能快速普及，两者相互促进。开源社区要有活力，核心是社区对开发者要有价值；它要能提供大量有益的项目、工具和模型，并为产业创造价值，这样一来，开发者会自发地积极贡献自己的力量，让社区持续繁荣。

2. 开源贡献

我参与了 TiDB 开源项目中一个简单函数的编写 `builtinNameConstDecimalSig`, 用于将 `NameConst` 向量化为十进制数。

函数块代码很少, 但是这次提交的 Pull request 让我了解到了如何完整地参与一次开源项目的开发: 包括提交源代码、进行自我测试、签署 CLA 协议、等待审核等一系列步骤。最后接收到 Merged 通知时自己也很高兴, 自己也为一款软件作出了微小的贡献。

<https://github.com/pingcap/tidb/pull/13214>

3. 课程反馈

在介绍 GitHub 的相关操作, 比如 fork 或者提交 pr 的时候, 希望可以建立一个用于练习相关操作的测试项目。例如一个简单的图书管理系统或者教务系统, 源代码用大家熟悉的语言编写, 适当的留出一些功能让大家共同开发; 大家也可以向这个项目添加一些更为丰富的功能。这样通过一些简单的“教程”帮助大家理解共同协作开发项目的过程, 进而让大家参与业界实际的开源项目, 减少同学们有参与开源项目的热情却不知道如何下手的窘境。

4. 参考文献

[1]张盖伦. 开源开放是人工智能发展主要趋势之一[N]. 科技日报,2019-10-17(003).