开源报告

骆绪锐 51195100044

1 课程总结

总结

毫不夸张说，本学期的《开源》课是我最喜欢的课。

除了一次去外面当志愿者之外，我没有缺过一次课。课上老师提到过的网站如「Hacker News」，书如「人月神话」等，有的成了每日必看，有的已经读完准备有空再看一遍，还有的也放在那里，等着有机会一定要去看看……

我觉得这门课，可能是整个研究生课程体系里，给我的收获最大的。

课程从开源、GitHub讲起，但不局限于此。传统的高校课程中，理工科多数就是在培养一个专业技能。并且由于考核指标、授课方式等原因，导致学生即使认真学，拿了高分，也并不代表其专业水平足够，可以在真实行业中立足。而开源课并不是一门讲技术，让在座同学们成为某个行当技能大牛的课。

它更多是一门关于分享的课，让很多人来到同学们面前分享自己的见解，给同学们更多的是启发，播下一粒粒种子。老师们没有手把手教，而是“授人以鱼不如授人以渔”，直接将经过磨练后精挑细选的方法论、网站、论坛等分享给我们。每打开一个，对很多人来说都是打开一扇新世界的大门，收获的全部都是惊喜。

传统课堂上，老师们对于技术更多是强调严肃认真地对待，而《开源》课里，更多会提起”玩”这

个字，是在享受这些有趣的技术，这一点对我来说非常契合。我也是一个热爱技术的geek，跨专业保研来读计算机。

最喜欢的课

很多嘉宾的分享都非常宝贵，尤其对在校生来说。我们平时里接触的，大多是老师和同学，对互联网的认知也大都是在网上，获取的信息都有很大的局限性和误导性。而那些行业里的“老炮儿”过来做分享，说的都是一线互联网江湖的种种是非故事内幕，是我们不知道翻遍多少个论坛、网站，请多少学长学姐吃饭才能了解到的。

我印象最深，最喜欢的一次嘉宾分享是庄表伟老师，也是我们的华师老学长，带来的分享《世界观与方法论》。他把自己的人生感悟用哲学化的形式与我们讨论。

在演讲里，最让我感到脊背发凉的是那句

> 成功的团队兼具多样性与平衡性，而且往往容易让一个超级领导者脱颖而出。

正如著名的「二八定律」所描述的那样，任何组织里，只有那么一小部分人，他们显示出来的价值是远远高于其它同伴的。

我就有切身体会。自己目前在实验室里和几个同学一起做一个项目，每个人都负责项目中的一些任务，几个人做成后的工作合起来就是一个完整的项目了。但是项目开始了几个月，潜移默化中每个人的角色，或者说重要程度就发生了改变。有的同学可能是之前来得早，在相关领域扎根深的缘故，就不知不觉成了组织领导者，被老师委以重任。而剩下的同学，尽管技术等层面也不逊色，但是会觉得项目存在感低，即使干了不少活，似乎最后都不会。

这时候，正如庄老师之后所言，应当趁早意识到，并且可以尝试去找一个自己为核心的新项目。这个方案，是逃离那个定律一个解。

演讲结束后，我主动向老师提问，“如果看待程序员到了35岁被裁？”，这个问题那段时间非常火。

在华为的庄老师迅速纠正到这是个谣言。他谈到两点，一是由于绩效只给30%的人打优，但是可能指标的缺陷无法能辨别出最优秀的人，因此打了B以下的有很多也是精英。那么他们选择出走是很明智的选择，因为市场会给他一个公道的报酬。二是程序员工作讲究「成长」或者说「生产力」，而要达到这个目的，是必须有「活力」才可以的，因此不能陷入加班的陷进，让自己没有思考、提高的空间。

建议

想给出的建议就是在介绍Git、GitHub这几个核心知识后，尽全力网络互联网大佬们来做现场分

享，讲自己的故事和感悟。

这些东西对同学们来说可能最重要，特别是以后的发展。并且往往这些人来了之后，来的人也会变多，大家也都听得十分认真。

2 项目总结

项目简介

我参与的项目是「Dubbo」，它已经较为成熟，并成功毕业，成为了Apache的顶级项目。

Apache Dubbo 是一个高性能的，基于Java的开源RPC框架，用于治理分布式架构。

第一次PR

PR链接

尽管之前也用过Github，不过更多的是传自己的项目，或者从Explore上Fork自己感兴趣的项目、教程等。对于贡献一无所知。

后来，在项目导师，北纬老师的引导下，我开始从了解项目做起。首先打开了中文主页，浏览官方文档，对整个dubbo项目有了一个整体的认识。

很快，第一次可以贡献项目的机会。我发现中文文档一处明显的错误，便第一次尝试提交PR。

1. Fork Dubbo-Website 项目
2. git clone 到本地
3. 修改相应的markdown文件
4. git add 该文件
5. git commit -m '修改文档'
6. git push origin master
7. 在自己的项目主页点「New Pull Request」
8. 再加上相应的备注，为什么要改，怎么改的

大功告成！

第一次做，花了我很长时间，还曾一度以为不需要fork就能贡…最后多亏了问自己的实验室学长。

最后合并成功，并且收到「LGTM」(Look Good to Me)后，我真的非常的兴奋激动。

学会单元测试

随后，在老师的建议下，我开始给项目做单元测试，提高项目的测试覆盖率。

可是单元测试又是什么鬼，我又没有系统学过，只是听说见过而已。

因此，我又去学了怎么用Junit插件来做单元测试。与此同时，又重新认识了比如Maven对开发的意义等，又花了几天时间，摸爬滚打后终于知道了单元测试是怎么回事。说白了，就是看一下你写的某个或几个方法到底能不能运行，你就单独给他们相应的参数，让他们单独跑一下就行了。

可惜的是，我没在测试方面提交PR，做出贡献。因为时间有限，实验室也有项目要开发，不得不停滞下来。可是，我把单元测试的技术成功用来了实验室项目的Java开发中。每写好一个方法，我再也不用单独配一个main函数，自己遍参数，然后编译运行了。自己写项目的效率得到大大提高。可见，努力都是有收获的。

第二次PR

PR链接

尽管没有在单元测试方面有贡献，我还是把dubbo的相关知识翻了个底朝天。并且会参与到各种网上的讨论里。

在知乎的一个帖子中，我发现题主有和我一样的困惑。正是由于dubbo主页的解释有歧义，才造成了很多新人对于「垂直应用架构」的理解有误。

再看看英文文档，发现里面最关键的「one way」没有翻译出来，正因为此，才造成了困惑。

我马上提交了PR，问题得到解决。

正是这个PR的提交，让我认识到官方网站，特别是官方文档对一个项目的重要性。同时，我也亲身体验到社区的意义，正是开源，才能让每个人成为项目的参与者，可以最及时、最有效解决大家头疼的问题。比公司里内部，层级管理的效率高太多倍。