

《开源软件设计与开发》课程总结

姓名：丁光耀

学号：52195100003

1、开源理解

结合本学期课程中学习到开源社区知识，我对如何建立一个开源社区兴趣颇深。通过自己查找的一些资料，谈谈如何建立开源社区。

社区就是有共同兴趣的一群人。开源项目和闭源项目都有用户社区，大部分用户不会积极地与社区其他成员互动。而另一方面，无论是开源社区还是闭源社区，都会有成员愿意更积极地参与，例如，报告 bug、帮助其他用户、撰写文档或进行推广。最活跃的社区成员甚至可以得到回报。例如，微软通过最有价值专家 (MVP) 项目奖励帮助人们充分利用微软技术的用户社区成员。在开源社区中，活跃的成员往往会获得额外的项目访问权和管理权。

虽然闭源项目也有奖励，但通过奖励社区成员而取得项目贡献毕竟有明确限制。由于不能公开验证代码，因此用户无法真正深入项目、解决问题或开发新功能，或者贡献代码。相反，在开源社区中，信息（代码和文档）可以从任何社区成员流入社区中心，尽管管理者会加以调整。更重要的是，一旦出现任何问题，社区中会有很多“眼球”盯着它，以便集思广益、群策群力。在闭源项目中，解决某个问题的最多人总是由受雇的开发者人数决定的。

社区创建之初可能非常小，可能只有一两个开发者，几乎没有用户。根据项目的类型，这种情况可能会持续一段时间，甚至几年，在此“孵化期”，初始团队会努力工作打下基础。Eric Raymond 在《大教堂与集市》中指出，成功的一个必要条件是，开发出可运行、可测试的东西，让游戏继续下去。注意，“可运行”并不意味着完美，甚至可以不完整。“早发布，多发布”是一条众所周知的开源开发准则，主要是因为这样做可以获得宝贵的早期反馈，帮助项目团队建立自信。因此，社区不应过早发布存在顾虑或迟疑；只要能够明确、如实地管理期望值，早期发布将使项目受益良多。

软件要最终吸引用户，软件的演示和品牌推广必须让潜在用户相信，该软件与竞争性产品相比有显著优势。引起用户的兴趣后，必须要降低进入的门槛：例如，安装等简单操作必须非常顺畅。用户注册并不是故事的结尾；从长远来看，还需要更多开发者的参与，他们至少可以完成少量工作，如果仅依赖个别开发者，他很可能被这些琐碎的工作压垮。开发者可能直接从用户中产生，但也可能来自其他地方，如为了挑战技术难题、为了声誉，或为提高或展示自己的编程水平而参与到项目中。

以这样的方式开放项目会带来新的复杂度。Karl Fogel 在《开发开源软件》一书中指出，“开放意味着以完全陌生的人可以理解的方式编写代码，建立开发网站和邮件列表，并往往需要撰写文档初稿。所有这些是很大的工作量。当然，如果真的出现了感兴趣的开发者，在他们作出贡献之前，还需要一段时间来回答他们的问题，这也是额外的负担。”而 GitHub、Google Code、StackExchange 等协作工具至少可以部分分担这些额外工作。此外，开放项目并不一定意味着失去控制权。许多项目在初期采用贤明君主模式，由一个人负责开发主要新功能并审核贡献的代码。

贤明君主的技术水平不必最强，但 Fogel 认为，他们应该“有辨别好的设计的能力”。但 Fogel 进而指出，他们的主要职责是确保参与者始终相信“团队合作胜过单打独斗”。团队领袖必须有足够的凝聚力，让开发者愿意持续参与项目，只有这样，开发者才会留下来。这意味着，应适当回报努力工作的人，如给予适当的肯定，或根据开发者的意愿赋予更重要的工

作职责。人们已将开源项目管理描述为一项活跃、灵活而低调的工作。

围绕开源软件建立社区可能是一项缓慢而艰巨的工作，其成功取决于很多因素。然而，没有社区也就没有项目。社区不会自动建立，社区的建立需要悉心管理。所有社区都始于被软件的包装、品牌推广或口碑推荐所吸引用户。一旦有了用户，不辜负用户的高期望又成为新的挑战。一个繁荣的开发者社区能够满足并超越这些期望，但前提是他的领导者具有凝聚力，能确保参与者不偏离轨道。从长远来看，社区需要落实开放开发机制，以确保包括创始人在内的关键贡献者离开时，他们的角色可由合适的人选填补。

2、开源贡献

本学期本人对开源所做贡献较少，但是接触到不少开源软件，如 Hadoop、Spark、Flink 等并详细了解其架构原理，希望以后能多为开源项目作出贡献

3、课程反馈

本次课程让我受益良多，老师对课程的规划也较为精细，特别是一些大牛的亲身经历和经验，能够带给人启发少走弯路。

4、参考文献

[1] Eric S-Raymond. 大教堂与集市[M].机械工业出版社,2014:5-10.