

《开源软件设计与开发》课程总结

武朝阳 51195100017

1. 开源理解

我认为参加开源社区，参与开源项目对于开发者个人的提升有很大的帮助。

- 开源可以提升代码质量：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，这个很可能会是你的主要动机。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。

开源工具当中我觉得对自己帮助最大的就是 Git 了，Git 的优点很多，但是这里只列出我认为非常突出的几点。

- 由于是分布式，所有本地库包含了远程库的所有内容。
- 优秀的分支模型，打分支以及合并分支，机器方便。
- 快速，在这个时间就是金钱的时代，Git 由于代码都在本地，打分支和合并分支机器快速，使用个 SVN 的能深刻体会到这种优势。

虽然有这么优秀的版本管理工具，但是我们面对版本管理的时候，依然有非常大得挑战，我们都知道大家工作在同一个仓库上，那么彼此的代码协作必然带来很多问题和挑战，如下：

- 如何开始一个 Feature 的开发，而不影响别的 Feature？ 由于很容易创建新分支，分支多了如何管理，时间久了，如何知道每个分支是干什么的？
- 哪些分支已经合并回了主干？
- 如何进行 Release 的管理？ 开始一个 Release 的时候如何冻结 Feature，如何在 Prepare Release 的时候，开发人员可以继续开发新的功能？
- 线上代码出 Bug 了，如何快速修复？ 而且修复的代码要包含到开发人员的分支以及下一个 Release？

大部分开发人员现在使用 Git 就只是用三个甚至两个分支，一个是 Master，一个是 Develop，还有一个是基于 Develop 打得各种分支。这个在小项目规模的时候还勉强可以支撑，因为很多人做项目就只有一个 Release，但是人员一多，而且项目周期一长就会出现各种问题。

Git 作为一个源码管理系统，不可避免涉及到多人协作。 协作必须有一个规范的工作流程，让大家有效地合作，使得项目井井有条地发展下去。"工作流

程"在英语里, 叫做"workflow"或者"flow", 原意是水流, 比喻项目像水流那样, 顺畅、自然地向前流动, 不会发生冲击、对撞、甚至漩涡。

现在广泛使用的工作流程有 Git flow 和 Github flow

Git flow 的优点是清晰可控, 缺点是相对复杂, 需要同时维护两个长期分支。大多数工具都将 master 当作默认分支, 可是开发是在 develop 分支进行的, 这导致经常要切换分支, 非常烦人。

更大问题在于, 这个模式是基于"版本发布"的, 目标是一段时期以后产出一个新版本。但是, 很多网站项目是"持续发布", 代码一有变动, 就部署一次。这时, master 分支和 develop 分支的差别不大, 没必要维护两个长期分支。

Github flow 的最大优点就是简单, 对于"持续发布"的产品, 可以说是最合适的流程。

问题在于它的假设: master 分支的更新与产品的发布是一致的。也就是说, master 分支的最新代码, 默认就是当前的线上代码。

可是, 有些时候并非如此, 代码合并进入 master 分支, 并不代表它就能立刻发布。比如, 苹果商店的 APP 提交审核以后, 等一段时间才能上架。这时, 如果还有新的代码提交, master 分支就会与刚发布的版本不一致。另一个例子是, 有些公司有发布窗口, 只有指定时间才能发布, 这也会导致线上版本落后于 master 分支。

上面这种情况, 只有 master 一个主分支就不够用了。通常, 你不得不在 master 分支以外, 另外新建一个 production 分支跟踪线上版本。

2. 开源贡献

- 自己在 Github 上开源了几个自己做的小项目，也有了一些 star
- 为一些开源项目修改过 bug

3. 课程反馈

上了这一学期的课程我感觉收获颇多，自己从一个不知道开源是什么的小白，变成了开源运动的倡导者，甚至贡献者。以后会更加努力地宣传开源精神，让社区变得更加壮大。