

《开源软件开发技术》2020 年作业 (2)

班级 大数据 1901 学号 2019011215 姓名 赵洁

1. 开源开发有哪些人员角色?

开源开发项目中的基本角色有：项目管理人员，开发人员，开发辅助人员，用户小组

- 项目管理人员的作用：指导项目的开展，组织协调，吸收性的特性和缺陷修正，软件的发布；开发人员的作用：主要参与软件的实现，文档的编写；开发辅助人员的作用：引入新的特性，报告缺陷，其他贡献；用户小组的作用有兴趣的客户，报告缺陷，建议新的特性。

开源开发项目中常见的角色有：维护者、贡献者、修订者。

- 维护者的作用：提交项目成果，维护项目运行，进行项目宣发，撰写项目文档，对项目方向负责，不断改进项目内容、水平；贡献者的作用：为项目做出有价值的贡献、想法、代码等等；修订者的作用：有目的性的将一个开源项目具化为某一方面的应用。

开源项目组织结构中的角色有：项目管理委员会（PMC），技术领袖，开发者（编码人员、评审人员），测试人员（代码测试、网站维护），赞助者。

- 项目管理委员会（PMC）管理开发者和测试人员，为开发者、测试人员、技术领袖和赞助者负责；开发者（编码人员、评审人员）与技术领袖和测试人员（代码测试、网站维护）交流沟通；测试人员（代码测试、网站维护）与赞助者和开发者（编码人员、评审人员）交流沟通。

2. 创建和开展开源项目需要注意的问题有哪些?

问题：

- 真的需要一个新项目吗？
- 特色与优势在哪里？
- 项目目标是什么（What? How? When?）
- 这个项目可行吗？
- 自己这个项目适合开源模式吗？
- 除了自己还会有人对这个项目感兴趣吗？
- 别人是否也有类似想法，有了雷同的项目？

需要注意的地方：

项目计划：

- 明确的目标，选择一个好的项目名称
- 人员组织结构规划
- 开发流程规划：
 - 新功能的开发流程
 - 缺陷报告、修复及发布流程
 - 代码提交及审查流程
 - 新版本发布流程
- 版本计划：版本编号办法
 - 近期发布的版本信息（包括功能列表、缺陷修复列表）
 - 长远规划
- 具体版本的开发计划
- 用户管理计划

- 推广计划
- 选择正确的许可证

在具体操作中需要注意的地方：

- 最初发布。当明确初步计划之后，应该将代码和设计公诸于众。如果你想让人们对你的工作感兴趣，第一印象至关重要。
- 这一阶段需要精心建立起归档的文件结构，例如，以项目名称为根目录，src 子目录存放源代码、docs 子目录存放文档，bin 子目录存放编译后的可执行代码等等，并建立 makefile、README 和 CHANGELOG 等文件。
- 按正确的原则来命名和准备文档。
- 项目上网。将代码公之于众有多种途径，例如可以通过 GitHub 或 SourceForge，也可以通过新闻组或建立自己的网站。发布前一定要仔细确认所有的程序都能正常工作。如果建立自己的网站，要有如下内容：
 - 项目的特点（为何要有这个项目，谁会对此项目感兴趣）
 - 版本控制
 - 缺陷追踪
 - 下载项目源代码的地方
 - 邮件列表、BBS
 - 常见问题解答（FAQ）列表
 - HTML 格式的项目文档
 - 相关的其他项目或网站的链接（BLOG,Wiki……）
 - 实时通信
- 开展合作。当你的文件发布当网上，人们开始知道你的想法之后，就应该设法去寻找有兴趣参与项目的合作伙伴。要让大家知道你要做什么，需要什么样的人，让每个人又能审视你的计划并提出意见。如有必要，修改计划以适合众人的想法。选用一种配置管理工具构建资源库，使大家的贡献能不断汇聚到一起。另外要照顾每人的特长，需要的时候把项目分解让每人做最适合做的事。
- 做好公关。要使项目让人感到有生命力，应该经常发布消息通报项目的进展情况，任何吸引人们注意、满足大家好奇心的事情都可以写进去，例如，哪些计划或哪部分代码发生了变化，下一发布版本的目标等等。也可以建立一个论坛或一个项目邮件列表。透过这个邮件列表，项目的合作开发者可以互相交流并讨论对程序修改的建议。
- 履行计划。别让人们为下一版本的发布期待过久，否则人们会怀疑项目是否进展不利。当然也别发布得太快，让人无法适应而引来众怒。要掌握好平衡，换言之，要合理计划，并使开发按计划进行。

3. 开源开发为什么要进行版本控制，如何进行版本控制？

原因：

- 开源开发往往需要不断的修改代码，当修改或删除项目代码时不希望破坏项目当前的状态，可以采用复制重命名原文件，在副本中进行操作的方法。当后面出现问题时可以快速找回文件之前的某个状态。
- 还有一个原因是最终发布软件时，源代码通常只有一份，只是在最后的编译阶段用到不同部分的项目文件，大部分项目文件的代码依然是相同、共用的。
- 版本控制的过程也是项目过程的追踪记录。

版本控制的方式：

- 集中式版本控制：代码集中到服务器，用的时候每个人把自己需要的被授权的那部

分代码下载到自己的计算机上，提交也是最后提交到服务器上，服务器可以对代码做很好的控制，但是需要有网络，网络断了，就没法工作了。代表就是 **SVN**, **Team Foundation** 等。

· 分布式版本控制：分布式自然就是每个人的地方多有一份完整的代码，提交和管理都是在本地进行，虽然有远端仓库，不过那是最终提交用的，没联网本地也是完整代码的，只是每次需要最新的代码的时候才必须联网。分布式的代表就是 **git**，对应的网上仓库比较有名的就是 **github**。

版本控制的注意事项：

不同的发布版本可能会带来兼容性的问题。新发布版本的某些文件可能无法与原先的某些文件共存于一个目录之中，这样应该考虑不同版本的安装目录用版本号加以区别。

4. 注册 Github 网站，填写个人资料（个人介绍，位置，个人网站等）。在 Github 创建一个仓库（repository），把本作业的文档上传到这个仓库中。将你的 Github 地址写到下面作为答案供检查。

<https://github.com/S217165/open-source>