《开源软件开发技术》2020年作业（2）

班级 大数据1901 学号 2019010284 姓名 倪少华

1. 开源开发有哪些人员角色？

**开源：**

1.维护者，**对于某些项目来说， “维护者”** 就是唯一拥有提交权限的人。然而在其它的一些项目中，他们只是自述文件中列出的维护人员。作为一名维护者，不一定非得一定要为项目撰写代码。有可能是项目的布道师，为项目的宣传做了很多的工作，又或者是撰写文档让更多的人参与进来。不管每天做什么，维护者就是那些对项目方向负责的人，并致力于项目的改进。

2.贡献者，**作为 “贡献者” 可以是任何人** ，只要提出issue或PR 就叫做贡献者，那些为项目作出有价值的都算（无论是分类问题，编写代码还是组织会议），又或者是将他们的PR合并进主干的。

3.修订者，**术语“修订者”**可能用于区分其他形式的贡献的提交访问，这是一种特定类型的责任。其实可以根据自己喜欢的方式来定义项目的角色，[考虑使用更广泛的定义](https://www.w3cschool.cn/opensourceguide/opensourceguide-prkf3bpr.html" \t "https://www.w3cschool.cn/opensourceguide/_blank)来鼓励更多的形式的贡献。无论技术技能如何，您都可以使用领导角色来正式识别为您的项目做出突出贡献的人员。

**开发：**

1.分析人员

业务分析人员的任务是理解和描绘客户的需求，引导和协调用户和业务需求的收集和确认，文档化和组织系统的需求，或者向整个团队传达需求。

2.架构师

架构师负责理解系统的业务需求，并创建合理、完善的系统体系架构。架构师也负责通过软件架构来决定主要的技术选择。这典型的包括识别和文档化系统的重要架构方面，包括系统的需求、设计、实现和部署"视图"。

3.数据设计人员

对于大多数的应用开发项目来说，用于持久存储数据的技术是关系型数据库。数据库架构师负责定义详细的数据库设计，包括表、索引、视图、约束、触发器、存储过程和其他的特定数据库用于存储、返回和删除持久性对象的结构。

4.项目经理

项目经理负责管理业务应用开发或者软件和系统开发项目。 项目经理角色计划、管理和分配资源，确定优先级，协调用户和客户的交互。项目经理也要建立一系列的实践活动以确保项目工作产品的完整性和质量。

5.开发人员

开发人员通常负责设计和实现可执行的代码方案、测试开发出了的组件和分析运行时情况以去除可能存在的错误。有时开发人员还负责创建软件的体系架构或者使用快速应用开发工具。

6.系统测试人员

系统测试人员负责制定测试计划并依照测试计划进行测试。这些测试包括功能性的测试(黑盒测试)和非功能性的测试(白盒测试)。测试人员需要良好的测试工具来辅助完成测试任务，自动化的测试工具将大幅度提高测试人员的工作效率和质量。

2. 创建和开展开源项目需要注意的问题有哪些？

**1.摘要**

如果你想发布一个开源库，请确保它有以下特点：

清晰的依赖性和安装说明

至少有一个简要的文档指南

修改日志和仓库中的标签

关于支持的语言、运行时、工具版本的信息和项目的成熟度

一个可以让用户提问和交流的邮件列表

**2.创建一个实用的README**

即使你的项目有一个很棒的网站，潜在的用户第一次接触这个项目很可能就是通过阅读README文件。我们需要确保它很棒并且包含了有用的信息。

**3.提供依赖信息**

清楚的说明项目的成熟度

运行时、语言、工具版本的文档支持

说明你使用了什么许可证

1. **为你的项目写文档**

写文档不容易同时也是需要花费一些时间的。然而，文档是你能为你的用户做的最好的事了。不仅能够节省他们大量的时间，也可以让他们确信你的库不是被遗弃的软件。

文档能够让你的用户完成他们起初使用你的库的任务。像Rob Pike说的，它“让这些任务成为可能”。这让你的用户知道你重视这一点，让他们知道你是个有血有肉的人，不是一个产生代码的机器。

1. **更容易升级**
2. **拥有一个修改日志**
3. **给版本加上标签**
4. **宣布版本发行**

在你发布一个版本字之后就是要写一个博客日志，或者在你们项目的邮件列表或更大的相关的邮件列表中发个更新。

1. **开发时使用预览或者快照版本**

在项目中用预览或者快照版本，当你快要发布一个版本的时候才揭开那个版本。然后****立即****升级那个版本。

**10提供一个让用户可以得到帮助的地方**

这可以让你创建一个社区，让你知道人们是怎么使用你的项目的、还有什么地方可以提高。最后，它会帮助你找到可以帮助你维护你项目的人。这不仅能节省你的时间，也会鼓励人们到处宣传你的项目。

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：**

主要是多个人合作的时候，可能会发生两个人同时提交一个文件的不同更改，这时候版本控制上就会出现冲突，就必须人工介入来决定保留谁的，否决谁的，当然这个事情是由仓库的建立者决定的。  
  目前的版本控制对文本内容有很好的支持，但是对二进制文件几乎没有支持，所以尽量不要在二进制文件上用这些版本控制工具，这不是他们的长处，二进制文件还是老老实实复制重命名吧。

**方法：**

**集中式版本控制**

  集中式顾名思义就是代码集中到服务器，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全的代码，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.

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/>