《开源软件开发技术》2020年作业（2）

软工1901 2019011635 王馨煜

1. 开源开发有哪些人员角色？

1): 项目管理员：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；

2）：开发人员：主要参与软件的实现，文档的编写；

3）：开发辅助人员 ：引入新的特性，报告缺陷，其他贡献 ；

4）：用户小组 ：有兴趣的客户，报告缺陷，建议新的特性。

2. 创建和开展开源项目需要注意的问题有哪些？

1）充分了解开源项目 ： 熟悉产品的功能、特点 ；

明确自己的定位 ；

熟悉代码规范与系统架构 ；

阅读与参与讨论 ；

2）参与社区的工作 ：关注邮件列表、新闻组、社区开发者大会，参与讨论

完善文档、文档翻译、测试并报告缺陷、提交补丁、 开发新功能、网站维护、项目管理

3）提升自己 ：提升技术水平 ；

保持热情 ；

在社区中树立良好声誉 ；

关注项目的发展 ；

尝试推广项目

4）邮件的礼仪规范：不要使用HTML格式撰写邮件 ；

客户端要支持邮件头属性“Reference”和“In-Reply-To” ；

使用UTF-8编码 ；

使用适合的邮箱地址 ；

尊重知识版权 ；

必要时忠实转发邮件 ；

不要骚扰别人（注意收件人列表和回复邮件时的抄送列表） ；

不要催人回复 ；

避免语言、宗教可能引起的误解 ；

正确使用大小写 ；

每行文字不要太长，以回车符换行 ；

回复邮件时避免“top-posting” ；

附件内容不要超过40KB ；

不要使用自动回复功能；

5）邮件列表的礼仪规范：需要时订阅和取消邮件列表 ；

向适合的邮件列表发信 ；

未经许可，不要随意公开邮件列表中的内容 ；

注意浏览已有的邮件列表 ；

反复斟酌发表的任何文字 ；

转发私人邮件前要获得对方许可 ；

发言应简明扼要 ；

使用标题，并应符合该邮件列表的习惯 ；

回复邮件时适当引用原文 ；

争论问题时对事不对人

6）提问的礼仪规范：社区乐于帮助别人解答问题，但鄙视提问前不愿思考，不愿付出的人 ；提问前，尝试通过手册、FAQ、邮件列表、网络搜索寻找答案 ；

详细、完整客观地描述问题的症状，以及发生的环境 ；

说清楚你要实现的目标 ；

对于代码中的问题，张贴代码时要加说明和注释 ；

解决问题后应告知问题是如何解决的，并向相关人员表示感谢

7）回答问题的礼仪规范：耐心 、启发式反问 、 授人以渔

8）讨论问题的礼仪规范：坦率地表达自己的观点 ；

提出建设性的意见 ；

只针对问题本身 ；

避免情绪化的措辞 ；

对待无礼行为的办法

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：

1. 最后发布软件的时候很多时候会有多个版本，而生成软件的源代码却往往只有一份，只是在最后编译生成的时候用到不同的部分，大部分代码还是共用的，所以往往需要版本控制，几个版本复制几个文件夹出来；
2. 没有版本控制系统的话，代码可能被别人或自己不小心覆盖或遗失，有了版本控制系统，开发人员只要将每次程式码的变更都纪录（Commit）起来，并且透过版本控制系统中进行更新。有了版本控制系统，可以浏览所有开发的历史纪录，掌握团队的开发进度，而且作任何修改都不再害怕，因为可以轻易的复原回之前正常的版本。

如何进行版本控制：

（1）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。

（2）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/>WXY8980/WXY-kaiyuan.git

https://github.com/WXY8980/WXY-kaiyuan.git