

## 《开源软件开发技术》2020 年作业（2）

班级 软工 1902 学号 2019013006 姓名 王亚伟

### 1. 开源开发有哪些人员角色？

维护者：可能拥有多重角色，可以写项目代码，也可以做项目的规划者，为项目进行宣传，或者编写文档，让越来越多的人参与进来，总之，维护者扮演者时时刻刻优化项目的角色。

贡献者：**作为“贡献者”可以是任何人**，只要他提出 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（或许这个定义是最接近所谓的贡献者的）。

**“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。

其实你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。

### 2. 创建和开展开源项目需要注意的问题有哪些？

确定开源项目（项目对外名称、logo、一般选取易于辨识的名称。）

2、确定托管平台。确定代码共享平台，GitHub 和码云 Gitee 都是目前使用率较高的代码托管平台，码云 Gitee 更适合中国的程序员开发。

3、选择许可证。启动开源项目前，必须确定使用的许可证、MIT、Apache2.0、GPLV3 都是目前使用率最高的开源许可证。GPL 不适用于商业软件；LGPL 适合作为第三方类商业应用。

4、制定项目说明。对你想开源的项目进行具体描述（项目介绍、应用场景、部署、开始、获取帮助、实现目标）。

5、编写贡献说明（提交错误报告的方式、如何建议新功能、配置环境的方式和运行测试）

6、建立行为标准。确立参与者行为规范，可以使用已有的准则比如贡献者盟约、Django 行为准备、Citizen 行为守则。

你要确定要公开的源代码都是你有权公开的。

选择合适的 license。特别是考虑是否要传染性的 license。

如果不确定自己是否侵犯了别人的专利，要采用有专利保护的 license。比如 GPLv3

### 3. 开源开发为什么要进行版本控制，如何进行版本控制？

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、也不知道是谁因为什么原因改了这段代码、也没办法可以复原回前几天的修改。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，我们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，因为你可以轻易的复原回之前正常的版本。我们也可以透过分支和标签的功能来进行软件发行的不同版本，例如稳定版本、维护版本和开发中版本。

由于源代码数量的急剧膨胀、变更的越来越频繁、可能修改同一个源文件的人也越来越多，需要将这些代码管理起来，于是每次变更被称之为一次修正（Revision）。版本控制更准确的说法应该是“Revision control”，每当我们修改一个源代码文件并再次保存时，就出现了两个不同的版本，一个是修改前的，一个是修改以后的。而版本控制，就是确保源文件的每一次修改，都被记录下来，并且可以知道是被谁修改的，是因为什么原因而修改的。必要时，可以找回任何一个版本的源代码。

版本控制工具：

仅管理本地源文件

- 免费/开源：SCCS (1972) RCS (1982)
- 闭源：PVCS (1985)

C/S 方式管理源代码

- 免费/开源：CVS (1990) CVSNT (1998) Subversion (2000)
- 闭源：Software Change Manager (1970s) ClearCase (1992) CMVC (1994) Visual SourceSafe (1994) Perforce (1995) StarTeam (1995) MKS Integrity (2001) AccuRev SCM (2002) SourceAnywhere (2003) SourceGear Vault (2003) Team Foundation Server (2005) Rational Team Concert (2008)

分布式管理源代码

- 免费/开源：GNU arch (2001) Darcs (2002) DVC (2002) SVK (2003) Monotone (2003) Codeville (2005) Git (2005) Mercurial (2005) Bazaar (2005) Fossil (2007) Veracity (2011)

- 闭源：TeamWare (1990s?) Code Co-op (1997) BitKeeper (1998) Plastic SCM (2006)

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/>\_\_\_\_\_