《开源软件开发技术》2020 年作业（2）

班级 计科 1904 学号 2019011283 姓名 黎浩

1. 开源开发有哪些人员角色?

维护者：

作为一名维护者，不一定非得一定要为项目撰写代码。Ta 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。不管他们每天做什么，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。

贡献者：

作为 “贡献者” 可以是任何人 ，只要 Ta 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）。

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?

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。

提供依赖信息

那么说你会发布你的开源项目。这说明你很聪明，真有你的！不幸的是，不是所有人都像你那样，而且有一些人对这门语言或者你在做的系统完全不了解。这意味着对你来说很显然的事情对他们来说就一点也不显然了。

其中一件就是缺少依赖说明或者安装说明

我到底怎么安装这个东西，难道不能说得清楚一些吗?

这很快就能让用户生气。在源代码里找你的包或者构件的名字是很烦人的，有些项目对构件使用一些特别有才的名字，完全不符合仓库的名字。

让你的用户从这些糟糕的事情中脱离出来吧。问题是怎样添加依赖性，理想状况下可以通过复制粘贴一小段代码。

如果需要例子的话可以点击 Welle。

清楚的说明项目的成熟度

你在生产中使用这个项目有几个月了吗？你是否觉得它还是不完整的？你是否希望 API 在下一个版本会彻底地修改？你的项目是否在要求最多并且很老的项目中也能稳定安全的使用？

要把这些说得清楚。下次你就不会因为做了一个错误的介绍，但是没有的提供任何项目成熟度的信息而项目浪费一周的时间了。你会意识到几句短短的话就能产生很大的影响。

运行时、语言、工具版本的文档支持

当考虑到向后兼容时，Clojure 有一个很好的跟踪记录。它好的几乎让人难以置信。包括 `1.2` 到 `1.3` 的升级，之后的升级对绝大多数的项目来说就是一个简单替换。同样地，那些高于 `1.2` 的项目大多使用了最新的稳定版本。

然而，不会一直都是这样。在某些情况下，未来版本的 Clojure 会打破兼容性。我们怎么让我们的用户不愤怒？通过在 README 中清楚的说明哪些版本是支持的。

这只需要写一行文字。这样，在你发布的那一周就少了抱怨，同时也减少了初学者的很多麻烦。

如果你需要一个例子，有一个来自 Welle 的例子。

说明你使用了什么许可证

你可能并不太关心许可证，但是那些在大公司中想用你的库的人很关心。他们必须知道！当他们想用 Clojure/Node.js/Scala/Go 等等的时候，可能不能使用。

因此清楚的说明你的许可证。也请你使用一些对商业友好的协议，除非你有自己的理由。(Apache Public License 2.0 和 Eclipse Public License）是不错的选择。注意到一些许可证(比如 MIT)的确很友好、流行，但是不提供任何专利保护，在当前的法律环境下也不应该忽视。

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。

3. 开源开发为什么要进行版本控制，如何进行版本控制?

（1）提高代码重用性。

（2）方便版本回退

（3）保存了数据当前状态以及之前每一个提交的历史状态,可以回退到任意一个版本节点;

（4）在保存每一个版本的文件信息时不重复保存数据，节约存储空间，提高运行效率;

（5）可以清楚到看到不同版本间修改的内容;

（6）可以多人协作，团队开发;

如何进行版本控制

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/_____