《开源软件开发技术》2020年作业（2）

班级 软工1903 学号 那婷婷 姓名 2019011150

1. 开源开发有哪些人员角色？

a. 软件架构师Software Architect: 主导系统全局分析设计和实施、负责软件构架和关键技术决策的角色。

b. 设计员Designer：在项目需求、构架和开发流程限制之下负责系统局部的分析设计的角色。

c. 实施员Implementer：遵照项目标准来开发、测试构件以集成为更大的子系统的角色。

d. 集成员Integrator：负责计划并执行集成活动从而将实施元素集成为产品构造的角色。

2. 创建和开展开源项目需要注意的问题有哪些？

1.完善文档，教用户如何使用。不要嫌弃用户。

2.响应用户的反馈，处理issue问题。创业产品的话就是要有客服体系了。

逐渐用户多了，然后形成社区，有了自己的品牌。

总结下这个过程中的关键点：

（1）DRY原则(Don’t repeat yourself)，不要重复自己，最常见的是用在代码规范里，建议大家不要随意复制粘贴，而是要做一定抽象。但实际上，所有的语言的高级功能，面对对象，模块化，代码生成工具，各种框架，都是在解决这个问题。也就是说，如果你发现你在做许多重复劳动，说明这里就有可能抽象出个工具出来。

（2）不要重新发明轮子。这个原则貌似有争议，但我觉得争议是没搞清楚“发明”和“造”的区别。不要重复发明轮子，但你可以造新的轮子，或者改进已有的轮子。这个原则说的是不要重复别人已经完成的工作，不要闭门造车，要在前人的基础上做改进。一直觉得发明轮子的人是很伟大的，也很难的，历史地位可以和发明取火术有的一拼。有了轮子后，人类所使用的工具和动物所使用的工具才有了本质的区别。

（3）前面我们做到了不要重复自己，也做到了不要重复别人，第三个境界就是“不要让别人重复你”。将你的工具，框架，抽象分享出去，作为开源产品或者SaaS服务，让别人不再重复你已经完成的工作。

2.开发速度慢了，竞品出现，或者功能比竞品弱。

3.推广没做好，大家不知道，结果被后来者超越了。

4.做出来的东西没有实际需求，比如pool，有人觉得go里用channel模拟就很简单，没必要用个复杂的pool。比如许多Copy To China的项目，发现在中国的环境，水土不服。

5.开源后不维护，没了热情，用户反馈无响应，客户都流失了。

3. 开源开发为什么要进行版本控制，如何进行版本控制？

版本控制的过程也是过程追踪记录，成就达成的过程。

原因：

1. 一个反复编辑，不厌其烦的编辑的过程。你修改的时候不希望破坏掉修改前的状态，最好就是每修改一段事件，能够保存一个状态，类似系统的自动快照一样，当后面出现问题的时候，可以自由选择复原到之前的某个快照状态。

2. 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。

3. 很多时候做一个项目，往往不是单枪匹马单打独斗的，是需要很多个人之间合作的，多人合作自然也需要进行多份文档可以编辑，最后再来合并。

方法：

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。代表就是SVN,Team Foundation等。

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/ntt-bistu>