

《开源软件开发技术》2020 年作业（2）

班级 计科 1906 学号 2019011178 姓名 刁昱文

1. 开源开发有哪些人员角色？

开发人员 Developers

主要包含以下角色：

- n 软件架构师 Software Architect
- n 设计员 Designer
- n 用户界面设计员 User-Interface Designer
- n 数据库设计员 Database Designer
- n 实施员 Implementer
- n 集成员 Integrator

角色

软件架构师 Software Architect

定义

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

职责

- I 领导与协调整个项目中的技术活动（分析、设计和实施等）
- I 推动主要的技术决策，并最终表达为软件构架
- I 确定和文档化系统的相对构架而言意义重大的方面，包括系统的需求、设计、实施和部署等“视图”
- I 确定设计元素的分组以及这些主要分组之间的接口
- I 为技术决策提供规则，平衡各类涉众的不同关注点，化解技术风险，并保证相关决定被有效的传达和贯彻

理解、评价并接收系统需求

评价和确认软件架构的实现

专业技能

技术全面、成熟练达、洞察力强、经验丰富，具备在缺乏完整信息、众多问题交织一团、模糊和矛盾的情况下，迅速抓住问题要害，并做出合理的关键决定的能力

具备战略性和前瞻性思维能力，善于把握全局，能够在更高抽象级别上进行思考；

对项目开发涉及的所有问题领域都有经验，包括彻底地理解项目需求，开展分析设计之类软件工程活动等

具备领导素质，以在各小组之间推进技术工作，并在项目压力下做出牢靠的关键决策

拥有优秀的沟通能力，用以进行说服、鼓励和指导等活动，并赢得项目成员的信任；

以目标导向和主动的方式来不带任何感情色彩地关注项目结果，构架师应当是项目背后的技术推动力，而非构想者或梦想家（追求完美）

精通构架设计的理论、实践和工具，并掌握多种参考构架、主要的可重用构架机制和模式（例如 J2EE 架构等）；

具备系统设计员的所有技能，但涉及面更广、抽象级别更高；

活动

确定用例或需求的优先级、进行构架分析、创建构架的概念验证原型、评估构架的概念验证原型的可行性、组织系统实施模型、描述系统分布结构、描述运行时刻构架、确定设计机制、确定设计元素、合并已有设计元素

工件

软件构架文档、参考构架、分析模型、设计模型、实施模型、部署模型、构架概念验证原型、接口、事件、信号与协议

角色

设计员 Designer

定义

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

职责

- | 理解、评价并接收系统需求细节
- | 理解、评价并接收相关的软件架构
- | 依据需求规格分析系统的内部行为，在分析层面识别与定义各系统组成元素的职责、操作
- | 识别与定义各设计元素的职责、操作、属性及其相互关系
- | 保证其设计符合软件构架，并足够详细到可以进入实施

专业技能

- | 掌握需求工程概念和技巧，以准确无误地理解相关系统需求；
- | 熟悉软件架构模式、概念和技巧，以准确无误地理解相关系统构架；
- | 精通软件设计理论、实践和工具，包括面向对象的分析设计技术和统一建模语言等
- | 掌握将用于实现系统的相关技术，例如组件开发（J2EE、EJB）、通讯机制、多线程与实时技术等；
- | 掌握将用于实现系统的程序设计语言（例如：Java、C++、C、HTML、CSS、XML、JavaScript、汇编语言）；
- | 对目标设计元素的相关问题有深入的了解
- | 熟悉项目的设计指南，明了设计与实施的关系，包括在实施之前设计应当达到的详细程度
- | 具备实施员的所有技能，但程度更深、抽象级别更高；
- | 通常兼任实施员的角色

活动

执行用例分析、设计用例实现、子系统设计、设计类、设计测试包与类库

工件

用例实现、分析类、设计子系统、设计包、设计类、测试类

角色

实施员 Implementer

定义

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

职责

- | 理解、评价并接收相关的软件设计元素
- | 在私有开发空间（workspace），遵照项目既定的标准（编程规范等）来开发、测试构件
- | 为了测试构件，需要相应的驱动和桩，实施员还负责开发、测试这些测试用构件及相关子系统
- | 将测试通过的实施元素（源码等）提交到集成空间，以交付给集成员来集成为更大的子系统
- | 重设开发基线（rebase），以便在新的基础上继续实施活动

专业技能

- | 熟悉软件设计模式、概念和技巧，以准确无误地理解相关设计；
- | 掌握软件编程理论、实践和工具，包括面向对象的编程技术和统一建模语言等
- | 精通将用于实现系统的程序设计语言和编程技能（例如：Java、C++、C、HTML、CSS、XML、JavaScript、汇编语言）；
- | 精通开发测试（特别是单元测试）和自动化测试工具，熟悉测试驱动编程更佳

- | 熟悉被实施构件的所有相关内容，包括其接口、结构与行为

- | 熟悉项目的编程指南

活动

实现开发测试（脚本等）、执行开发测试、实现设计元素、实现测试设计元素、分析运行时行为、编写构建脚本、开发安装工件

工件

开发测试（脚本等）、实施元素（源码与描述文档等）、构建脚本（makefile）、安装包

角色

集成员 Integrator

定义

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

职责

- | 计划在不同的子系统、系统级别下，进行构建和集成的顺序

- | 在集成空间执行集成，解决编译与链接冲突，调试构建脚本等

- | 建立工作基线

- | 将集成测试通过的子系统、系统实施元素提交到更

- | 上一级集成空间，以便集成为更大的系统

专业技能

- | 熟悉将用于实现系统的程序设计语言（例如：Java、C++、C、HTML、CSS、XML、JavaScript、汇编语言）；

- | 精通自动化构建工具（例如 make/ant）

- | 熟悉被集成子系统、系统的相关内容，包括其各组成部分间的依赖关系、与外

部系统间的依赖关系，以及它们的开发和依赖关系如何随时间而演变的预期

- I 具备优良的协调能力，以便与多个开发人员协作，来共同保证集成的成功

- I 很多时候可以兼任测试员的角色

活动

创建集成空间、计划子系统集成构建、计划系统集成构建、执行子系统集成、执行系统集成、建立基线、提升基线级别

工件

集成构建计划、构建脚本（makefile）、构造（build）

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，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，说明使用了什么许可证，为项目写文档，

3. 开源开发为什么要进行版本控制，如何进行版本控制？

因为修改程序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；

还有一个原因就是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；

版本控制的过程也是过程追踪记录，成就达成的过程。

以及很多时候做一个项目，往往不是单独的，而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

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/>_____