2019软工试题答案整理

多选题

- 云平台PaaS(中间层或平台层)层的作用
    - 对虚拟机池资源进行
        - 资源监测
        - 预警
        - 优化决策
- 遗留软件为什么要修改
    - 软件需要进行适应性调整，从而可以满足新的计算环境或者技术的需求
    - 软件必须升级以实现新的商业需求
    - 软件必须扩展以使之具有参与更多新的系统和数据库的互操作能力
    - 软件架构必须进行改建使之能不断适应不断演化的计算环境。
- 重构代码方法
- 某个原则
    - P90, 工作实践
- process 包括的内容
    - activity
    - action
    - task
- 软工定义
    - 将系统化的、规范的、可量化的方法应用于软件的开发、运行和维护
    - 对上面方法的研究
- process pattern包括的内容
    - 模式名称
    - 驱动力
    - 类型
        - 步骤模式
        - 任务模式
        - 阶段模式
    - 启动条件
    - 问题
    - 解决方案
    - 结果
    - 相关模式
    - 已知应用和实例

简答题

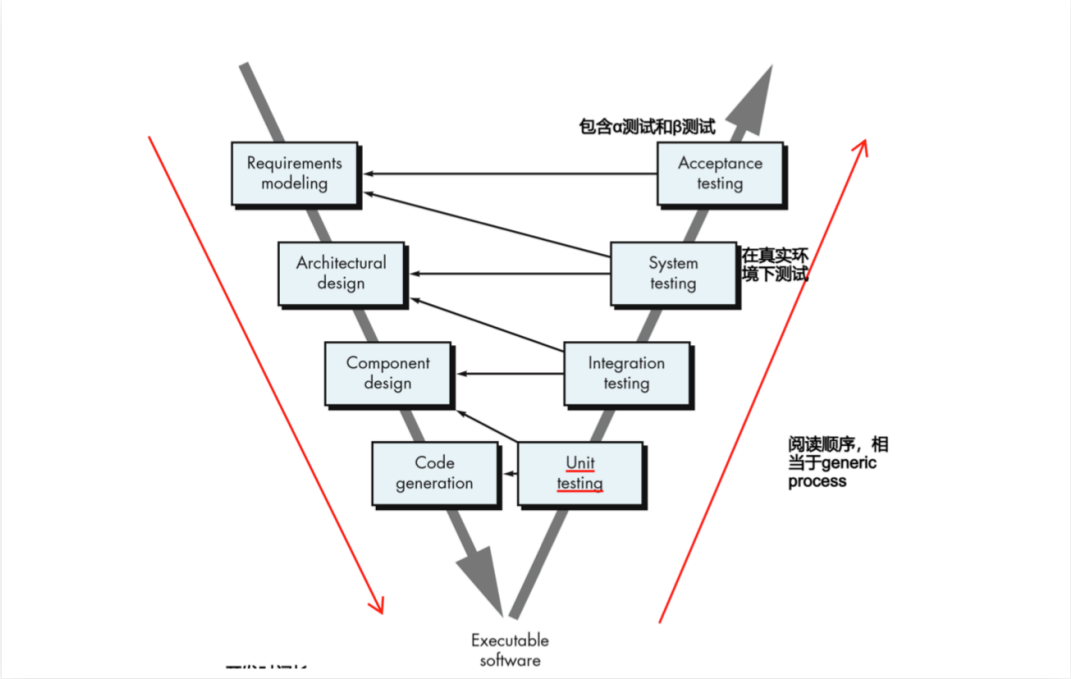
1. 举例描述Scrum开发全过程
    - 人员
        - Product Owner:杜老师
        - Scrum Master:邱博
        - Team:三个组员
    - 我们首先要确定一个Product Backlog(按优先顺序排列的一个产品需求列表)
    - 团队根据product backlog做工作量的估算
    - 通过Sprint Planning meeting, 来从中挑选出一个Story作为本次迭代完成的目标，时间周期是1~4个星期，然后把这个Story进行细化，形成一个Sprint Backlog
    - 在Sprint backlog再细化成更小的任务，成员领取任务(2天的工作量左右)
    - 过程中要进行每日站立会议，控制在15分钟左右，汇报昨天完成了什么，今天要做什么，遇到了哪些问题
    - 做到每日集成，每天都可以有一个成功编译，并且可以演示的版本。
    - 当一个Story完成，业绩就是Sprint backlog完成，表示一次Sprint的完成，这时，要进行Sprint review meeting。
    - 最后就是Spring Retrospective Meeting, 总结会议，每个人总结并讨论改进，放入到下一次Sprint的产品需求中。
2. 举例说明领域建模
    - 比如我做一个进销存系统，首先需要学习领域知识
        - 从已经开发过的项目获取
        - 通过用户调研获取
        - 调研目前和将来的需求
    - 然后进行领域建模
        - 通过需求规约里面的文字描述，把潜在类提取出来.
        - 对于一些通用的业务类，定义标准的方法和属性，然后以后做其他类型的进销存系统可以复用
        - 然后进行**类的功能建模**
            - 功能建模要构建activity diagram，反映类的功能逻辑：比如有一个订单类，客户产生订单成功了，就往订单加物品，如果没有成功，就返回尝试产生一个新的订单
        - 最后要选择一种建模语言进行建模，比如UML
3. 详细说明需求规约，需求分析规约，设计建模之间的关系
    - 建模和设计的关系
        - 场景建模
            - 用例图，用例描述
            - activity dia功能建模，可以用于整个**系统，用例，类，方法**
            - Swimlane dia:也是互动图，就是把参与活动的分析类标识出来，需要做这个功能需要哪些类参与
        - 行为建模
        - 数据建模
            - 类图
            - 类分析包，把功能相同的类放在一起作为一个包，比如一个界面包，处理报表包。
                - 什么叫子系统，不同子系统通过接口来访问。对外访问都通过接口类。比如生成报表，这就叫做package
            - CRC：告诉我们类图的时候与其他类合作的方法放进去，与组件设计有关
            - 协作图：类之间怎么协作的，为什么要协作，我这个类要调用其他类方法，得到信息，其他要调用我的方法，我告诉他信息，是定义类方法用的
            - 数据设计和类设计

        - 数据库设计(主要)
            - 以分析类图为依据，一对一，一对多，多对多关系相当与ER图里面的关系。
                - 如果类是一对一关系，则可以产生两个实体表，他们之间的关系可以放到任何一个实体里面去
                - 如果是一对多关系，生成两个表，关系放在多里面去
                - 如果是多对多关系，三个表，两个实体表，关系+实体属性的新表
        - 数据结构设计
        - 类设计
            - 表示对后台业务类进行设计和细化，例如加上其他合作者的方法，合并或拆分一些业务类
    - 体系结构设计
        - 设计类图中的后台那部分类是以分析类图为基础的
        - 架构设计包括两个方面
            - 三层
                - 跟分析无关，与环境有关，比如就是web系统，前端浏览器，后端服务器，就两层
            - 类之间调用关系
    - Component level design
        - 详细设计/组件设计：需要参考分析类图，因为需要设计类之间的相互调用，看调用关系然后增加方法。还需要依据状态图，例如打印机缺纸事件发生了，要调用打印机里面的缺纸方法来报警。到了缺纸状态怎么处理，看状态就知道就需要设计一个缺纸处理的方法。
4. 举例说明ACD在体系结构设计中的作用
    - 可以描述出所有的外部然健接口
5. 举例说明网页应用/移动应用中交互建模、功能建模、内容建模、导航建模之间的关系
    - **内容建模**要找到内容对象， 可能是文本，图片，video,声音,log。这些对象可能有树状的关系，按一下就到一个界面，然后到另一个界面，最后到叶子，最开始是root，最下层就是leaf。对象是嵌套的。
    - 内容对象放在一起，形成一个内容对象流，就是**导航建模**，完成一个特定的功能。
    - **交互建模**，首先要找到内容对象，不同的内容对象之间通过交互，完成特定功能。交互建模借用时序图， 每个时序图的生命线不是对象了，可以理解为界面层，
    - **功能建模**，要实现功能的逻辑是什么，例如退货，首先要找到订单，如果订单没有过期就可以退，首先要通过退货的活动图，把退货功能的流程图画出来，然后找到退货功能要涉及到哪些内容对象的交互。活动图又用到了。
6. 软件过程框架中定义了许多activities, actions, 和tasks, 举例说明并解释它们之间的关系
    - activity主要实现宽泛的目标，与应用领域，项目大小，结果复杂性或者实施软件工程的重要程度没有直接关系
    - 动作action包含了主要工作产品生产过程中的一系列任务
    - 任务task关注小而明确的目标，能够生产实际的产品

      举例
        - 例如testing这个activity,包括actions
            - 单元测试
            - 集成测试
            - 系统测试
            - 验收测试

              单元测试这个action包括tasks:
                - 计划
                - 用例分析

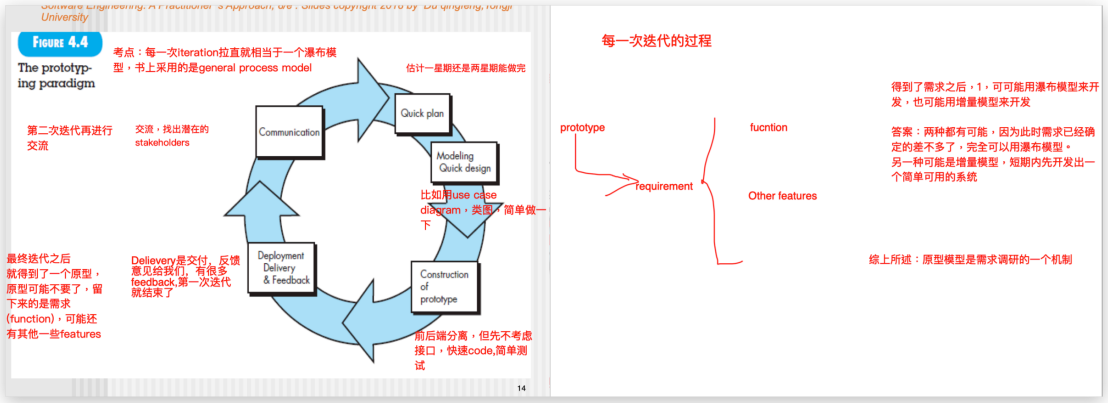
        - 设计测试用例
        - 正规技术评审
        - 运行测试用例
        - 修正bug
        - 软件配置管理
7. 软件配置管理
    - 软件配置管理是指一套管理软件开发和维护过程中所产生的各种中间软件产品的方法和规则。
    - 分支(branch)是开发的一条"支线"，独立于其他开发线路，并且和其他线路并行开发
    - Tag:通常tag对应于milestone, 是一个完整可用的版本，不能修改，是只读的。
8. 简述软件工程通用的过程框架包括哪5个活动，并说明活动之间的关系
    1. 沟通：包含了与客户之间大量的交流和协作，理解利益相关者的项目目标，并收集需求以定义软件的特性和功能
    2. 策划：指为后续的软件工程工作制定计划，它描述了需求执行的技术任务，可能的风险，资源需求，工作产品和工作进度
    3. 建模：包括创建模型和设计两个方面，创建模型有助于客户和开发人员更好地理解软件需求，设计可以实现需求
    4. 构建：包括编码和测试
    5. 部署：将软件交付到用户手中，并对其进行评测并给出反馈意见
9. 简述瀑布模型，简述瀑布模型与V模型的关系
    - 瀑布模型：又称为经典生命周期，它提出了一个系统的，顺序的软件开发方法，从用户需求规格说明开始，通过策划、建模、构建、部署的过程，最终提供完整的软件支持。
    - 瀑布模型的一个变体称为V模型



        - 随着软件团队工作沿着V模型左侧步骤向下推进，基本问题需求逐步细化，形成了对问题及解决方案的详尽且技术性的描述。一旦编码结束，团队沿着V模型右侧的步骤向上推进工作，其本质上是执行了一系列测试，与V模型没有本质区别。
10. 增量过程模型
    - 增量模型在每个阶段都运用线性序列。每个线性序列生产出软件的可交付增量
    - 第一个增量往往是核心产品
        - 例如开发一个文字处理软件
            - 第一个增量中提供基本的文件管理、编辑和文档生成功能
            - 第二个增量中提供更为复杂的编辑和文档生成功能

- 第三个增量中提供拼写和语法检查错误

11. 演化过程模型
    - 演化模型是迭代的过程模型
    - 原型开发泛型(帮助软件开发人员和利益相关者理解需求)



    - 沟通：找出潜在的利益相关者，明确并收集需求
    - 快速策划：估计一星期把原型做完
    - 快速建模：用use case diagram, 类图简单建模
    - 原型构建：前后端分离，先只做前端让用户能够看到界面
    - 部署：delievery是交付，反馈意见给我们，第一次迭代就结束了

    - 螺旋模型
        - 结合了原型的迭代性质和瀑布模型的可控性和系统性特点
        - 举例：
            - 比如说我做一个银联系统，成立了一个公司，然后中标，第一次个讨论得到一个结果，整理成初步文档，二三次讨论形成了一个需求规约草稿，交给银联领导，然后进行专家总评，得到一个正式的规约
            - 接下来要求在7，8个月内把原型做出来
            - 第四次迭代就做好了第一次原型，到了第五次原型，就形成了比较完整的需求规约
            - 在原型的基础上，成立架构设计组，数据库设计组等等,然后正式开发，测试，第六次，第八次迭代的时候要交付
            - 螺旋的每圈都会跨过策划区域，此时需根据交付后用户的反馈调整预算和进度，以及完成开发的迭代次数。

12. XP模型
    - Planning
        1. 评估每个story的优先级和价值
        2. 验收测试，根据用户提的stories来进行测试，提前订好测试的内容，比如要测试登录，购物，加入购物车，结算等业务
        3. 然后定下工作量和人员安排，迭代时间和交付时间
    - Design
        1. CRC:比如我要开发一个汽车零部件销售系统：这一次sprint有注册，订购，电话销售的功能，我要实现这三个功能，是部分的业务流，然后看有哪些类，零部件，供应商，用户这几个类，把他们属性确定下来，然后看他们怎么交互。形成一个CRC卡片，反应类之间是怎么交互的.
        2. Spike solutions prototypes:我在设计的时候（数据库的设计，接口的设计，类里面几个属性，几个方法），不确定哪一种好，遇到这种情况的时候，就要开发一个原型，是假的，针对这个功能开发一个原型，针对具体的业务来简单测试一下。然后确定到底用哪一种方案好。
    - Coding

        - 重构：写代码的过程中要持续对代码优化，重构。
        - Unit test:这次迭代有5个story, 我们实现了两个，产生了2个类，就可以对这两个类分别进行单元测试
        - Continuous testing:两个通过了单元测试的类可以和已经迭代完成的类集成起来测试
    - Test
        - 项目负责人对一些重要的类进行专门的测试，按照在planning阶段定下的acceptance test完整地对业务流进行测试
13. 云平台
    - 资源层或基础设施层(IaaS)
    - 中间件层或平台层(Pass)
        - 监测
        - 预警
        - 优化决策
    - 应用层或软件服务层(SaaS)