考试科目名称 软件过程与管理 （A卷）

考试方式： 闭卷 考试日期 2015 年11月 日 教师 荣国平

系（专业） 软件学院（软件工程） 年级 班级

学号 姓名 成绩

题号	一	二	三	四	五	六	七	八	九	十
分数										

注意：所有作答请写直接写在卷面上。

得分	

一、敏捷方法的特征有哪些？哪些关于敏捷特征的说法施加于敏捷方法之上是不合适的？为什么？（本题满分10分）

敏捷方法最主要他体现是小周期迭代式持续交付。遵循如下价值观：

- **敏捷软件开发宣言的4个简单的价值观：**
  - **个体和交互 胜过 过程和工具**
  - **可以工作的软件 胜过 面面俱到的文档**
  - **客户合作 胜过 合同谈判**
  - **响应变化 胜过 遵循计划**

关于敏捷方法的一些特征表述可能带着一顶的误导，例如：

轻量级方法：这是对以XP为代表的一类方法的误导，事实上，这类方法对工程规范有着极为严格的要求；

拥抱变更、变更驱动：仅仅是口号，对待变更，所有软件工程方法都是限制和管理的态度。

TDD可以提供更高的开发质量：并没有足够的证据支持。

得分	

二、产品组件集成策略有哪些？请解释这些策略的优缺点。在此基础上，解释如何结合质量指标来实现高质量集成？（本题满分10分）

大爆炸集成策略

该策略将所有已经完成的组件放在一起，进行一次集成。这是一种看起来非常具有吸引力策略。因为这有可能是需要测试用例最少的一种方式。然而，这需要所有待集成的产品组件都具有较高的质量水平，否则，难以定位缺陷位置的缺点会使得该策略消耗很多测试时间。而且，系统越复杂、规模越大，问题越突出。

逐一添加集成策略

该策略与上述的大爆炸集成策略完全相反，采取一次添加一个组件的方式进行集成。因此其优点就在于很容易定位缺陷的位置，特别在产品组件质量不高的情况下，每次集成之前都有着坚实的质量基础。但是，该方法的缺点也很突出。这可能是需要测试用例最多的一种策略，而且，大量的回归测试也会消耗很多时间。

集簇集成策略

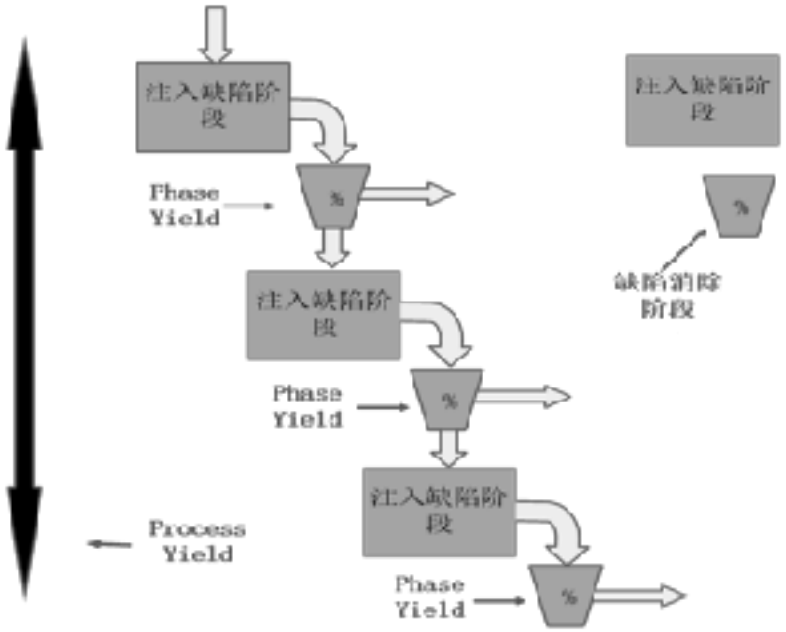
集簇集成策略是对逐一添加集成策略的改进。简单的随机选择产品组件进行集成并不合理。为了提升测试效率，往往会把有相似功能或者有关联的模块优先进行集成，形成可以工作的组件。然后以组件为单位继续较高层次的集成。此外，这种策略还有一个好处就是，可以尽早获得一些可以工作的组件，有利于其他组件测试工作的开展。但是，这种策略的缺点是过于关注个别组件，而缺乏系统的整体观，不能尽早发现系统层面的缺陷。

扁平化集成策略

该策略要求尽快构建一个可以工作的扁平化系统。也就是说，优先集成高层的部件，然后逐步将各个组件、模块的真正实现加入系统。这种方式可以尽早发现系统层面的缺陷。然而，该策略的缺陷是为了确保完成的系统，需要大量的打“桩”（*stub*），即提供一些直接提供返回值的伪实现。这种方式往往不能覆盖整个系统应该处理的多种状态。

得分	

三、请基于Yield度量指标构建缺陷预测模型，并列举该模型的可能改进方案。（本题满分10分）



由上图，只需要知道注入阶段缺陷注入水平（速度或者密度）以及消除阶段的缺陷消除水平（速度或者能力）就可以基于*yield*构建一个基本的缺陷预测模型。

可能的改进是假设注入水平和消除水平都符合正态分布，因此，可以用蒙特卡罗方法模拟结果。

得分	

四、谈谈你对项目估算（时间和规模）的认识（包括原理、方法和目标等），并简要解释应用PROBE 方法进行估算的优缺点。（本题满分10分）

规模估算往往可以依据历史数据来完成，其原因在于规模估算结果的偏差产生原因相对客观，偏差可以用以修正新的估算结果。

时间估算的偏差产生原因更加复杂，一方面和规模有关，另外一方面，跟人的主观能动性有关，因此，时间估算偏差的原因可能估算结果本身，这使得历史数据中时间偏差可参考价值不大。

从上述讨论可以得出，对于估算来说，本质上是一种猜测，追求的目标应该是一致性以及估算结果的使用者对估算结果的信心。

*PROBE*方法通过定义的估算过程和数据收集以及使用的框架，使得估算结果可以尽可能一致，从而使得一些统计方法可以用来调整估算结果，增强用户对估算结果的信心。

但是这种估算方法非常依赖高质量的历史数据，一旦数据不完整或者缺失，就可能导致估算结果有显著偏差。

得分	

五、如果对质量的追求是无止境的，在不考虑资源和成本的前提之下，有哪些可能有效的策略？（本题满分10分）

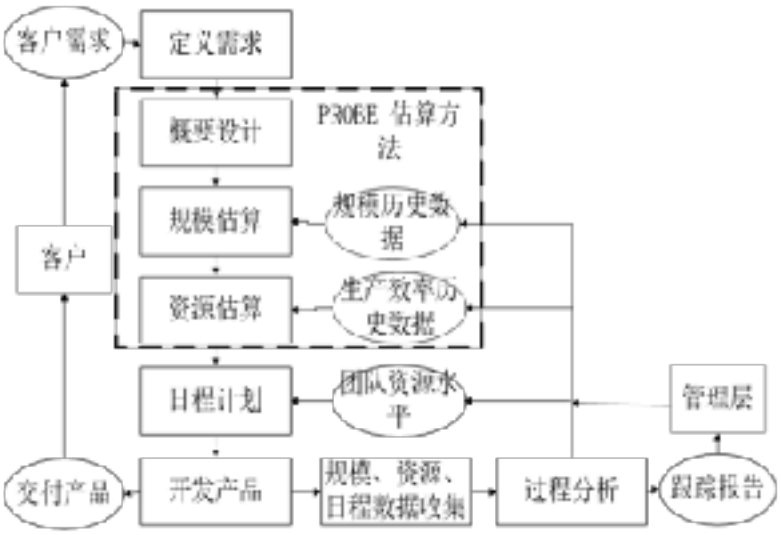
- ✓ 重视测试，并且将测试过程文档化并且稳定化；
- ✓ 重视小组评审，同样定义评审过程，并且使得评审过程的*performance*稳定化
- ✓ 重视个人评审，提升评审者技能；
- ✓ 重视设计
- ✓ 开展设计验证

得分	

六、如何开发一份让人无法拒绝的日程计划？请描述其基本步骤和一些注意事项。（本题满分10分）

这种日程计划的关键是必须用正推的方式来制订项目计划。一个典型的项目计划框架如下（*1*）：

（*8*）



在这个过程中，除了概要设计和资源估算之外其他环节尽可能自动化完成。充分参考历史数据进行估算。（*1*）

得分	

七、请列出Capture-recapture方法进行缺陷预测的假设条件和相应的模型定义（本题满分10分）

常见*CRC*模型定义了两个参数，即评审者发现缺陷能力*t*和缺陷的难度*h*，*t*是否一致以及*h*是否一致都会影响模型。一边来说，定义如下*4*个基本模型：

假设*h*和*t* 都一样的*M0*模型；

假设*h*不等而*t* 都一样的*Mh*模型；

假设*t*不等而*h*都一样的*Mt*模型；

假设*t*和*h*都不等的*Mth*模型

得分	

八、请结合软件开发的特点介绍软件项目管理中自主型团队的必要性（本题满分10分）

软件开发一种智力活动，开发者是智力劳动者，而对于智力劳动者而言，管理的第一准则就是智力劳动者不能被管理，只能实现自我管理：

- ✓ 处理和讨论非常抽象的概念
- ✓ 把不同的部分整合成一个可以工作的正确的系统
- ✓ 全身心地参与
- ✓ 努力做出卓越的工作

得分	

九、请描述CMMI模型的5个等级的特征，并且解释为何CMMI模型不应该是敏捷方法的对立面（本题满分10分）

（*1*分每个点）

1. *Initial* 原始级别，开发相对混乱，依赖个人英雄主义，没有过程概念，救火文化盛行；
2. *Managed* 已经管理级别，项目小组级体现着项目管理的特征，有项目计划和跟踪，需求管理、配置管理等等；
3. *Defined*，已经定义级别，在公司层有标准流程和相应的裁剪规范，每个项目小组可以据此定义自己的过程，使得优秀的做法可以在公司层共享；
4. *Quantitatively Managed,* 定量管理，构建预测模型，已统计过程控制的手段来管理过程项目；
5. *Optimizing* 持续优化，继续应用统计方法识别过程偏差，找到问题根源并消除，避免未来继续发生类似问题。

（*5*分）

*CMMI*模型本身并不是开发模型，而是一个过程改进的模型，刻画了软件组织从不成熟到成熟的路线图，简单说，*CMMI*模型不是一种具体的开发方法。而大部分所谓的敏捷方法都是开发方法，因此，两者是完全不同性质的事物，将这两者对立是不合适的。

得分	

十、请结合A/FR、PQI、Review Rate、DRL、Yield尽可能具体描述一个软件项目应该如何从多方面来确保开发的高质量（本题满分10分）

这些指标既是开发过程中质量管理的一些参考指标，同时也体现在计划安排中应该注意的质量元素。具体如下：

1. 在项目计划过程中应该安排确保高质量开发结果的活动，例如，按照*A/FR*、*PQI*等指标的要求，安排对各类产物（文档和代码）的个人评审和小组评审；
2. 这些评审活动应该满足一定的要求，特别体现在时间方面。例如，评审时间应该多于测试时间的两倍以上（*A/FR*）；评审时间应该是相应开放时间的*50%*以上（*PQI*）；评审速度要求（*Review Rate*）等
3. 充分借鉴质量指标所体现的开发质量状况，尽早制订相应的质量补救措施。例如，*PQI*所体现的缺陷密度、所有上述指标的参考值等等。一旦超标，往往意味着质量方面有偏差，应当及时补救。
4. 利用*yield*等指标，构建质量预测模型，更加积极（*Proactive*）地判定和控制开发质量；
5. 依据*PQI*和*Yield*指标所体现的信息，通过过程改进来提升开发质量。