Chapter 5 敏捷开发Agile Development

一种传统软件工程项目的合理的替代品

简述：敏捷软件工程可以快速提供成功的系统（重要特性：**适应需求变更**）

强调：

- 注重"客户的满意度"和"尽早提交可增量的软件产品"（增量交付，开发过程存在迭代）
- 快速交付，不看重中间产品
- 小型，积极的项目团队，协调能力强，看中团队结构，协作态度
- 可用非正式的方法（e.g. 数据库设计没有写完整的设计文档，中间建模尽量简单）
- 最小的work product（开发过程简单，会要求尽量减少work product且work product简单，不完全按照template）
- 整体发展简单（e.g. 封闭开发）
- 软件工程师和其他项目利益相关者（管理人员，客户，最终用户）在一个敏捷的团队，命运共同体，其中roles和responsibility随时调整

特点：

1. story / function： 通过需求调研得到项目的一些功能(story / functions)
2. priority：对story设置优先级

   在这之后，选择优先级高的stories进行第一次迭代，得到可演示产品；demo后进行第二次迭代，包括对上一次的stories进行修改，加入新的stories,……
3. 每次迭代完后均是可以演示运行的产品

基本框架活动 CPMCD (generic process,当然也可以遵循standard process)

5.1 Agility 敏捷

正在构建的软件的变化，对团队成员进行更改，由于新技术而更改，可能对其构建产品或创建产品的项目可能产生影响的各种类型。

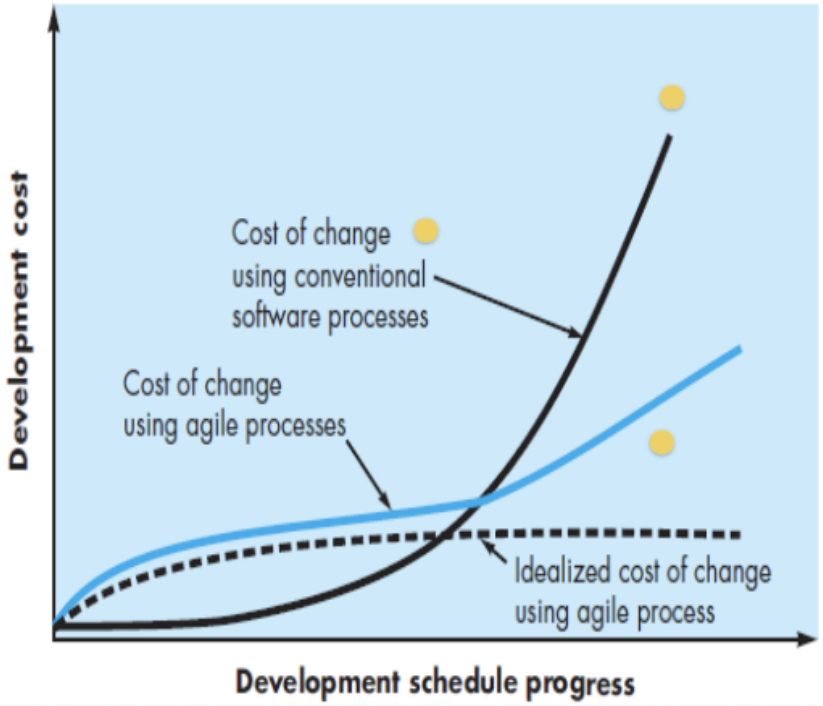
5.2 Agility 和 Change Cost 改变成本

※ Change Cost图(出选择题，要理解)

FIGURE 5.1

Change costs as a function of time in development

回归测试：2000(add)+1000(modified)+600(delete)，则总共的3600个用例进行regression testing



理解传统software processes的黑色曲线：

比较显然，随着软件开发逐步变化，需求变更可能需要改的部分越多（例如：还在modelling阶段和coding阶段change cost完全不同，coding阶段cost会大得多）

理解敏捷模型的曲线：

regression testing回归测试——变更多了后，回归测试成本越来越高，最终翘了上去（这一版和上一版本的修改会有牵连——相关用例）

回归测试：

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。自动回归测试将大幅降低系统测试、维护升级等阶段的成本。

回归测试作为软件生命周期的一个组成部分，在整个软件测试过程中占有很大的工作量比重，软件开发的各个阶段都会进行多次回归测试。在渐进和快速迭代开发中，新版本的连续发布使回归测试进行的更加频繁，而在极端编程方法中，更是要求每天都进行若干次回归测试。因此，通过选择正确的回归测试策略来改进回归测试的效率和有效性是很有意义的。

5.3 敏捷过程(Agile Process)

敏捷开发解决的问题：

1. 没有办法提前预测需求是什么，会怎么改，优先级如何
2. design和construction有交叉，过程中的activity是串联的。这样比较难去预测多少design在construction之前是必要的
3. 分析analysis, 设计design, 构建construction和测试testing和预想的不同

敏捷过程的目的：创建一个可以对付不可预知性的process(to rapidly changing project)

结论：需要一个增量的开发战略increment development strategy

5.3.1 敏捷原则Principles (12个)

※ 考试会出选择题（加混淆的选项）

1. **Highest** 最高优先级：通过早期early和连续continuous交付有价值的软件来满足客户
2. **Change** 鼓励需求改动，哪怕是开发过程晚期（好的改动可以提升顾客的竞争优势）
3. **Frequently** 经常提供work product，最好有短的schedule
4. **Together** 甲方和开发者daily紧密合作
5. **Motivated** 和有动力的人开发，给他们好的环境和支持，信任他们
6. **F-to-F** 面对面讨论，实时解决问题，这是最高效率的
7. **WorkingSoftware** 进度的最好指标就是产出的可工作的软件
8. **Sustainable** 敏捷过程保持稳定的开发过程，赞助人，开发者和用户应该maintain a constant pace indefinitely (不定期同步跟踪)
9. **Excellence** 持续关注追求技术的改进和设计的优化
10. **Simplicity** 文档尽量简单
11. **Self-organizing** 一个self-organizing的team可以做出最好的架构、需求和设计
12. **Reflection** 定期进行回顾

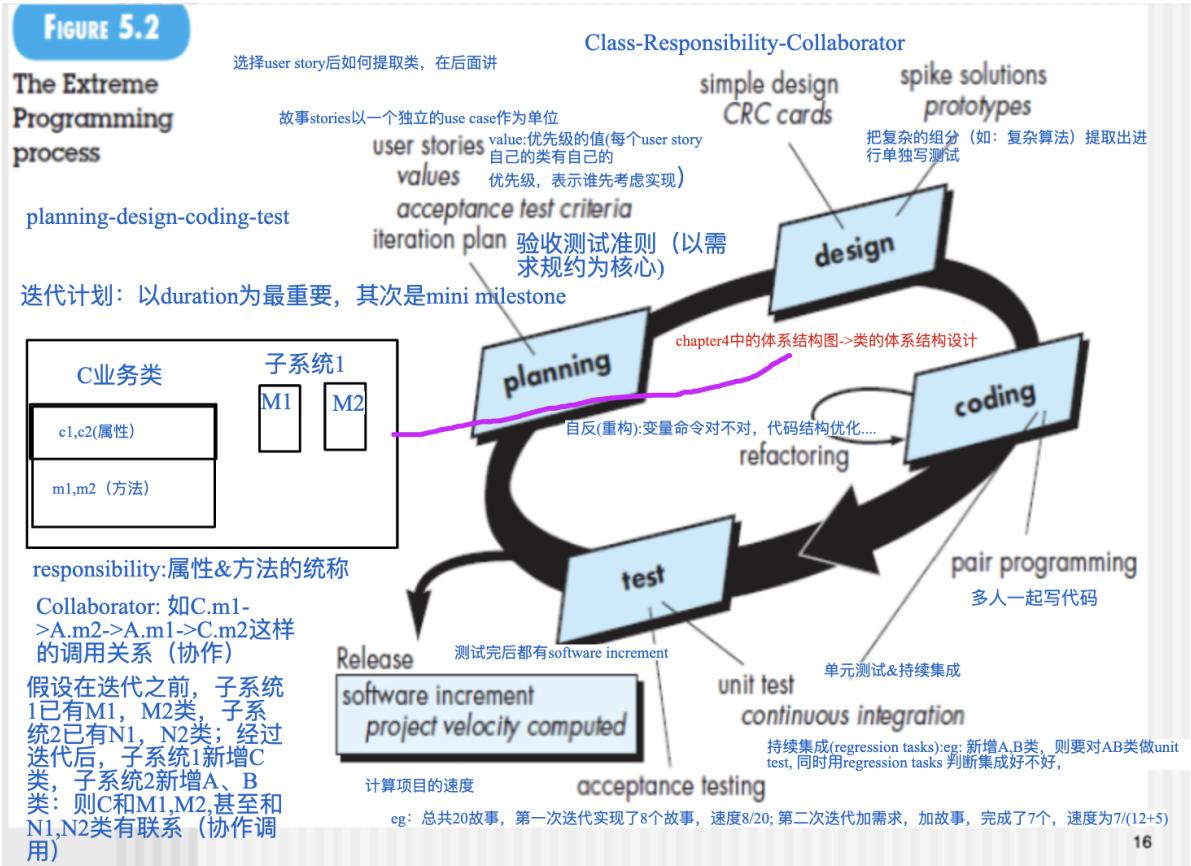
5.4 极限编程Extreme Programming

5.4.1 XP Process

是agile model

是迭代开发的，每轮结束后都可以运行的系统

4个framework activities：Planning，Design，Coding和Testing（区别Generic Process Activity的Framework(CPMCD)



- Planning: 导出需求
    1. 设定一系列stories，故事的单位是use case

2. 客户基于特征或功能的整体业务价值为story分配优先级，划分依据：
    1. 业务上重要性
    2. R此功能(story)是否有高风险(high risk)
    3. D交付时间(deadline要求)
3. 如果估计故事需要超过三个发展周(工作量太大），请客户拆分为较小的故事
4. 一旦基本承诺定下（关于要包括故事的协议agreement，交货日期和其他项目问题），XP团队会将以三种方式之一开发story：
    1. 所有story将被实施立即（几周之内）
    2. 最高优先级的story将在计划中提前并首先实施
    3. 最高风险的story将在计划中提前并首先实施
5. 第一个project（software increment）在发布后，XP team计算目前完成的速率(project velocity)——第一个release版本中部署的customer stories的数量。它可以用于评估接下来release的发布日期和开发schedule，以及看整个项目的所有stories是否存在overcommitment(过度承诺)，如果有，则要修改release内容或推迟delivery dates
6. 迭代

- Design：
    - 鼓励使用CRC卡片(Chapter 10)
    - **Spike Solution**——快速解决问题的模型一种Design Prototype，对没有把握的部分先给出一个快速解决方案（原型）在特定的环境下跑一下（有时不知道是否正确，时间复杂度等也未知，此时可以先快速开发，看有没有把握去解决问题）
- Coding

  设计完成后，团队不会转移到代码，而是开发一系列unit test，该测试将测试当前版本中包含的每个story（软件增量software increment）：业务流程的梳理

  1. Pair Programming 结对编程（互相看逻辑做Check）
  2. Refactoring 重构（对代码规范化，标准化以进行复用）
  3. Unit test

     在programming前首先将单元测试的测试用例写出来，用test case覆盖逻辑

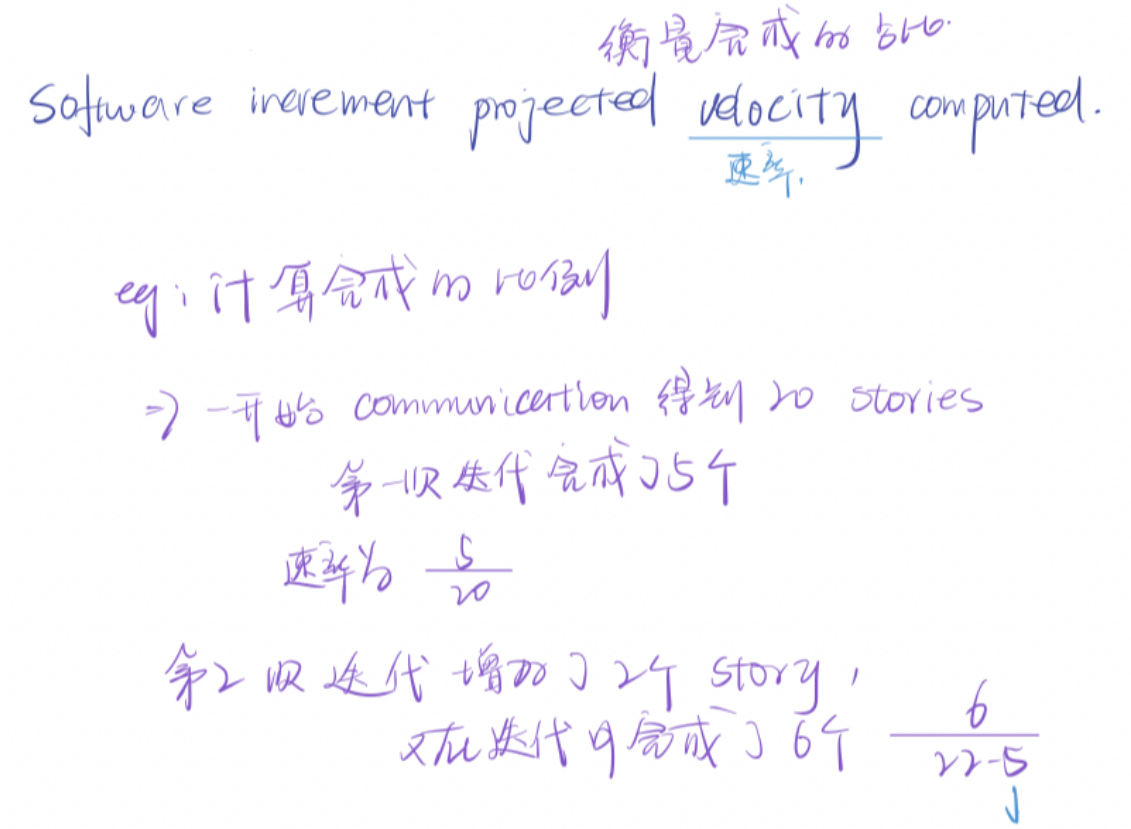
     pair programming后马上就运行unit test的测试用例（测试先行）

     - Continuous integration 2个人结对编程时把两个人的代码做基础

  （注意！continuous integration的目的是把两个人的代码做集成；而test中的continuous integration是把不同的pair program写的代码做集成）

    - Test
        - Continuous integration 将pair program结对编程小组的代码整合
        - Unit test 需重新运行coding阶段的test case；再加一些test case（集成中发现的问题，因为一般一些测试类用例是优化顺序的
        - 做成一个test suite测试套件，生成test case set测试集正常执行
        - acceptance test: 一般是开发人员模拟用户将story的逻辑全部测试一遍
- Release:

  software increment project velocity要计算，必入衡量完成的占比



敏捷原则：

1. 个体与交互胜过过程与工具
2. 可以工作的软件胜过面面俱到的文档
3. 重视客户协作（每次迭代的完成）
4. 响应变化

5.6 SCRUM

一种敏捷开发框架，是增量迭代的开发过程（很重要，考试必考SCRUM或XP）

阅读公盘的课外材料