

软件工程

第五章 敏捷开发

徐本柱 软件学院

2018-09

敏捷软件工程

- ❖ 哲学理念和一系列开发指南的综合。
- ❖ 哲学理念
 - ❖ 推崇让客户满意和软件尽早增量发布；
 - ❖ 小而高度自主的项目团队；
 - ❖ 非正式的方法；
 - ❖ 最小化软件工程项目以及整体精简开发。
- ❖ 开发方法强调
 - ❖ 超越设计和分析（尽管并不排斥这类活动）的发布
 - ❖ 开发人员和客户之间主动和持续的沟通。

敏捷开发

- ❖ 敏捷开发恰当的称呼应当是“类软件工程”
- ❖ 保留了基本框架活动：
 - ❖ 客户沟通、策划、建模、构建和部署，
 - ❖ 缩减到项目组朝构建和交付发展的最小任务集
- ❖ 客户和软件工程师有共同的观点：
 - 唯一真正重要的工作产品是在合适时间
 - 提交给客户的可运行软件增量。
- ❖ 敏捷团队认为过程可行，开发出的可交付软件增量能使客户满意，则敏捷方法已经正确实施。

敏捷宣言



♥ 个人和互动	胜过	流程和工具
♥ 工作软件	胜过	宽泛文档
♥ 客户协作	胜过	合同谈判
♥ 变化响应	胜过	计划遵循

右边各项很有价值，左边各项价值**更高**

敏捷方法

- ♥ 敏捷方法是为了克服传统软件工程中认识和实践的弱点开发而成的。
- ♥ 不确定性意味着**变更**，而变更意味着付出**昂贵的成本**，特别在其失去控制或疏于管理的情况下。
- ♥ 敏捷方法最具强制性的特点之一就是它能够通过软件过程来**降低由变更所引起的代价**。

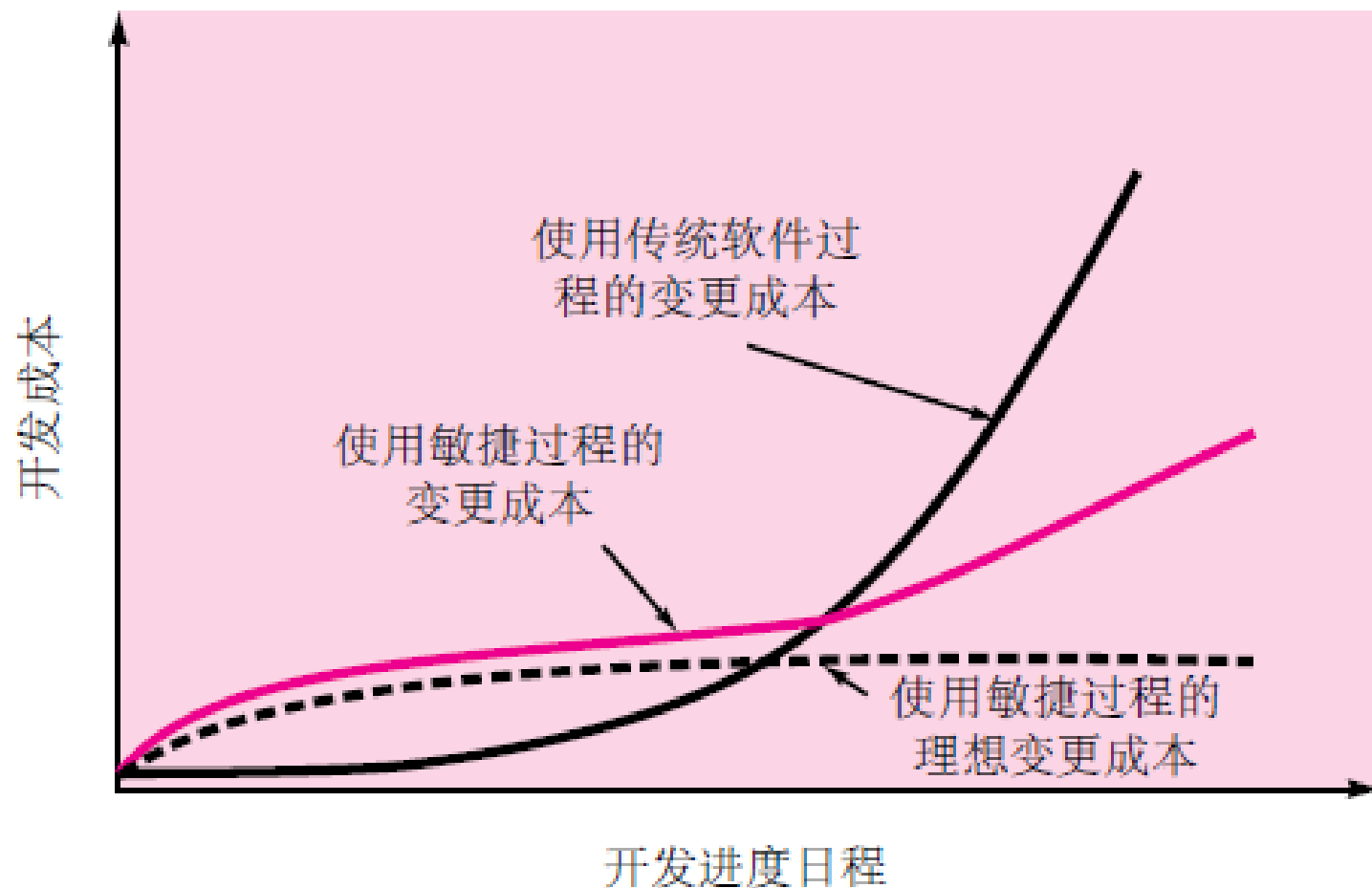
- ❖ 忘记了开发计算机软件的人员的弱点
 - ❖ 软件工程师在工作方式上有很大差别，
 - ❖ 在技能水平、主动性、服从性、一致性和责任心方面也有巨大差异。
 - ❖ 一部分人可以通过书面方式很好地沟通，而有些人则不行。
- ❖ 过程模型 “利用纪律或者宽容来处理弱点”
 - ❖ 大多数惯例过程模型选择了纪律，
 - ❖ “不能一致连贯地做同一件事是人性的弱点，因而高度纪律性的方法学非常脆弱。”



5.1 什么是敏捷

- ❖敏捷——描述现代软件过程的时髦用词
- ❖有效地（快速、灵活）响应变化
- ❖利益相关者（经理、客户、最终用户）间的有效沟通
- ❖将客户作为开发团队的一部分
- ❖组建高度自主的项目团队
- 最重要的是...*
- ❖快速交付给客户可运行的软件增量

5.2 敏捷变更成本



5.3 敏捷过程

❖ 三个关键假设:

- ❖ 1. 提前预测哪些需求是稳定的而哪些需求会变更非常困难。预测项目进行中客户优先级的变更也很困难。
(不可预测)
- ❖ 2. 设计和构建是交错进行的。(增量开发)
- ❖ 3. 分析、设计、构建和测试并不像我们所设想的那么容易预测。(迭代、反馈)
- ❖ 重要问题: 如何建立能解决不可预测性的过程?
 - ❖ 过程的可适应性——快速变化的项目和技术条件
 - ❖ 即敏捷过程必须具有自适应性
 - ❖ 原地踏步式的连续适应性变化收效甚微——敏捷软件过程必须增量地适应

敏捷过程

- ♥ 由用户所需的应用场景驱动
- ♥ 认识到计划时间很短
- ♥ 使用增量式开发策略
- ♥ 交付多个软件增量版本
- ♥ 能做出适应性变更

5.3.1 敏捷联盟定义的12条原则

- 1、我们最优先要做的是通过**尽早、持续交付有价值的软件**来使客户满意。
- 2、即使在开发的后期，也**欢迎需求变更**。敏捷过程利用变更为客户创造竞争优势。
- 3、经常交付可工作软件，交付的间隔可以从几个星期到几个月，交付的时间间隔**越短越好**。
- 4、在整个项目开发期间，**业务人员和开发人员**必须天天都**在一起工作**。

敏捷联盟定义的12条原则（续1）

- 5、围绕受激励的**个人**构建项目。给他们提供所需的环境和支持，并且信任他们能够完成工作。
- 6、在团队内部，最富有效果和效率的信息传递方法是**面对面交谈**。
- 7、可工作软件是**进度**的首要度量标准。
- 8、敏捷过程提倡**可持续的开发速度**。责任人、开发者和用户应该能够保持一种长期、稳定的开发速度。

敏捷联盟定义的12条原则（续2）

- 9、不断地关注优秀的技能和好的设计会增强敏捷能力。
- 10、简单（使不必做的工作最大化的艺术）是必要的。
- 11、最好的架构、需求和设计出自于自组织团队
- 12、每隔一定时间，团队会反省如何才能更有效地工作，并相应调整自己的行为。

5.3.2 敏捷开发战略

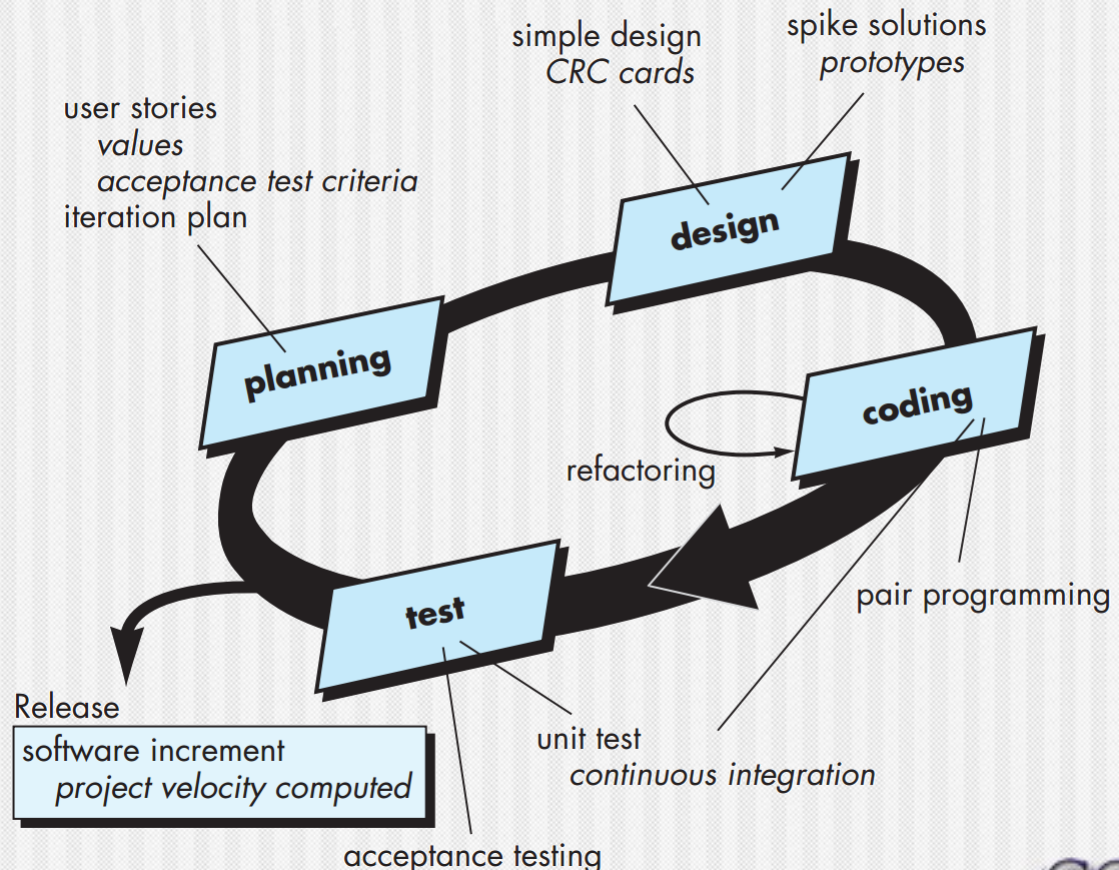
- ♥ 与传统方法在优越性和适用性方面存在争论：
 - ❖👍传统方法乐于产生完美的文档而不是可运行的系统
 - ❖👎自以为了不起的黑客，妄图将玩具软件→企业级
- ♥ 真正的问题在于“什么是最佳实现途径”？
- ♥ 如何构建满足用户当前需要的软件，同时展示具有能满足客户长期需求的扩展能力？
- ♥ 没有绝对的答案
- ♥ 敏捷学派内部也存在多种模型

敏捷过程模型

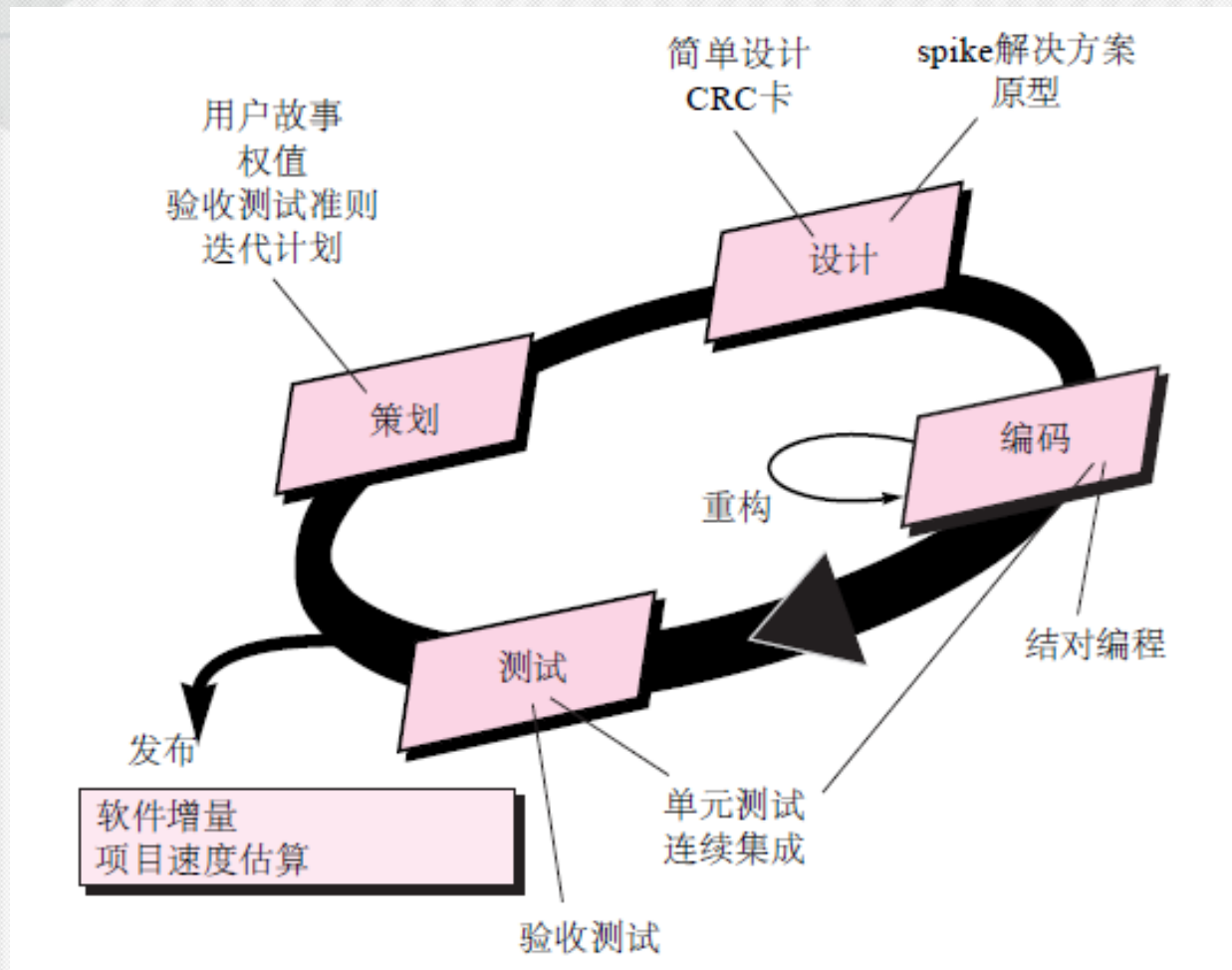
- ❖ 敏捷方法（或多或少地）都遵循敏捷软件开发宣言以及敏捷原则。
- ❖ 常见的敏捷过程模型有
 - ❖ 极限编程、自适应软件开发、动态系统开发方法、Scrum、Crystal、特征驱动开发等。

5.4 极限编程XP

- ❖ XP使用面向对象方法作为推荐的开发范型。
- ❖ XP包含了**策划**、**设计**、**编码**和**测试**4个框架活动的规则和实践。
- ❖ 如图



极限编程



极限编程过程

极限编程——策划

♥ 使用最广泛的敏捷过程，由Kent Beck提出。

♥ XP 策划

- ❖ 开始于倾听，倾听产生一系列“**用户故事**”。
- ❖ 团队成员评估每一个故事，并给出以开发周数为度量单位的**成本**。
- ❖ 团队共同决定如何将故事分组，并置于将要开发的下一个**软件增量**中。
- ❖ 给出对下一个发布版本的基本承诺（就包括的故事、交付日期和其他项目事项）
- ❖ 项目的第一个发行版本（也称为一个软件增量）交付之后，XP团队计算**项目的速度**，用于帮助估计后续发行版本的发布日期和进度安排。

极限编程——设计

❖ 设计：

- ❖ XP设计严格遵循KIS原则，使用简单的表述。
- ❖ 为故事提供**恰到好处**（不多也不少）的设计，不鼓励额外功能性设计。
- ❖ 鼓励使用**CRC卡**（第9章）
- ❖ 如果设计中碰到困难，推荐“**Spike解决方案**”——一种设计原型。
- ❖ 鼓励“**重构**”——以不改变代码外部行为而改进其内部结构的方式来修改软件系统的过程。

极限编程——编码和测试

❖ 编码：

- ❖ 推荐在编码开始之前建立单元测试（测试驱动）
- ❖ 鼓励结对编程。

❖ 测试：

- ❖ 所有单元测试应当使用一个可以自动实施的框架。
 - ❖ 支持重构和回归测试
- ❖ “验收测试”由客户规定技术条件，并且着眼于客户可见的系统级特征和功能

工业极限编程IXP

- ❖ IXP是XP的一种有机进化。
 - ❖ 由XP最低限要求、以客户为中心、测试驱动精神组成
- ❖ 主要差别在于其管理具有更大的包容性
 - 扩大了用户角色，升级了技术实践
- ❖ IXP合并了六个新实践
 - ❖ 准备评估
 - ❖ 项目社区
 - ❖ 项目承租
 - ❖ 测试驱动管理
 - ❖ 回顾
 - ❖ 持续学习

5.5 其他敏捷过程

- ♥ Scrum
- ♥ 动态系统开发方法
- ♥ 敏捷建模
- ♥ 敏捷统一过程等

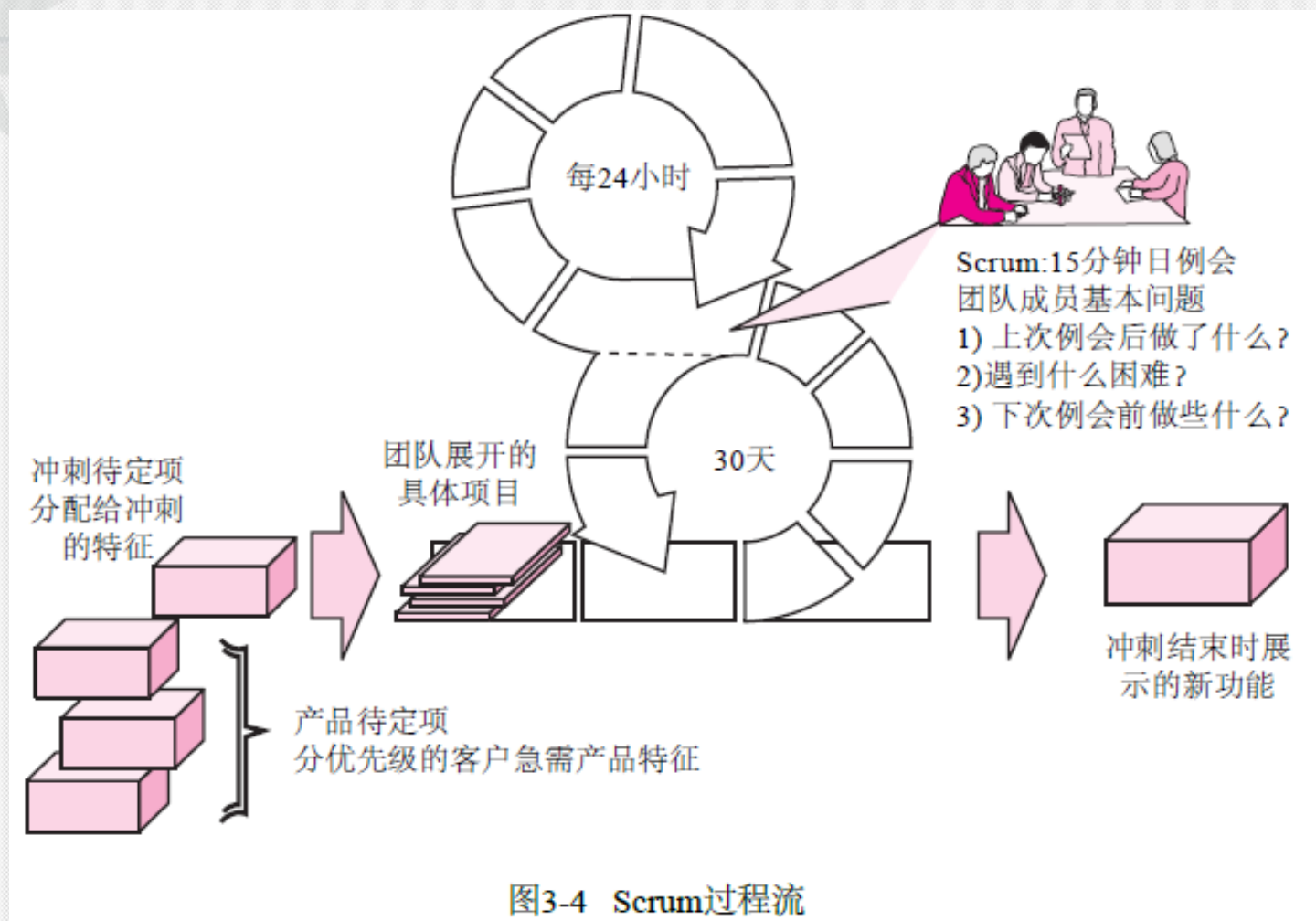
Scrum

- ♥ 由Schwaber and Beedle提出
- ♥ 应用Scrum原则指导过程中的开发活动
 - ❖ 过程由“需求、分析、设计、演化和交互”等框架活动组成
 - ❖ 发生于一个过程模式中的工作任务称为一个冲刺(sprint)，冲刺中进行的工作常常实时修改
 - ❖ 例会时间很短，有时甚至站立开会
 - ❖ 在规定时间内将演示软件交付给用户

Scrum

- ❖ **Scrum** 强调使用一组“软件过程模式”，这些过程模式被证实在时间紧张的需求变化的和业务关键的项目中是有效的。
- ❖ **待定项(backlog)**：一个能为用户提供商业价值的项目需求或特征的优先级列表。
- ❖ **冲刺(sprint)**：由一些工作单元组成，这些工作单元是达到待定项中定义的需求所必需的，并且必须在预定的时间段内完成。冲刺过程不允许有变更。
- ❖ **Scrum例会**：**Scrum**团队每天召开的短会。

Scrum



动态系统开发方法(DSDM)

- ❖ 一种敏捷软件开发方法，提供一种框架
- ❖ “通过在可控项目环境中使用增量原型开发模式以完全满足对时间有约束的系统的构建和维护”
- ❖ 交付整个系统需要100%时间，
 - ❖ 80%的应用系统可以用20%的时间交付

动态系统开发方法(DSDM)

- ❖ 以建立基本的业务需求和约束的可行性研究开始
- ❖ 识别功能和信息需求的业务研究
- ❖ 定义了3个不同的迭代周期：
 - ❖ 功能模型迭代——为客户开发一系列可证明其功能的增量原型
 - ❖ 设计和构建迭代——功能模型迭代中，重新构建原型确保工程化方式实现，提供可操作的业务价值
 - ❖ 实现——将最终软件增量置于运行环境中实现
 - ❖ 增量不一定100%实现
 - ❖ 增量置于运行环境以后可能需要变更

敏捷建模

- ♥ 由Scott Ambler提出
- ♥ 提出一系列的敏捷建模原则
 - ❖ 有目的的建模
 - ❖ 使用多个模型
 - ❖ 轻装上阵
 - ❖ 内容重于表述形式
 - ❖ 理解模型及工具
 - ❖ 适应本地需要

敏捷统一过程AUP

- ❖ 每个AUP迭代执行以下活动：
- ❖ 建模
- ❖ 实现
- ❖ 测试
- ❖ 部署
- ❖ 配置及项目管理
- ❖ 环境管理

5.6 敏捷工具集

- ♥ 自动软件工具
- ♥ 协作和沟通工具
- ♥ 项目管理工具
- ♥ 。 。 。



谢谢!

