
第5章 敏捷开发

AGILE DEVELOPMENT

敏捷软件开发的宣言

“我们正在通过亲身实践以及帮助他人实践的方式来揭示更好的软件开发之路，通过这项工作，我们认识到：

- 个人和他们之间的交流胜过了开发过程和工具
- 可运行的软件胜过了宽泛的文档
- 客户合作胜过了合同谈判
- 对变更的良好响应胜过了按部就班地遵循计划

也就是说，虽然上述右项很有价值，但我们认为左项具有更大的价值。”

个体和互动 高于 流程和工具
工作的软件 高于 详尽的文档
客户合作 高于 合同谈判
响应变化 高于 遵循计划

*Kent Beck et al,
2001*

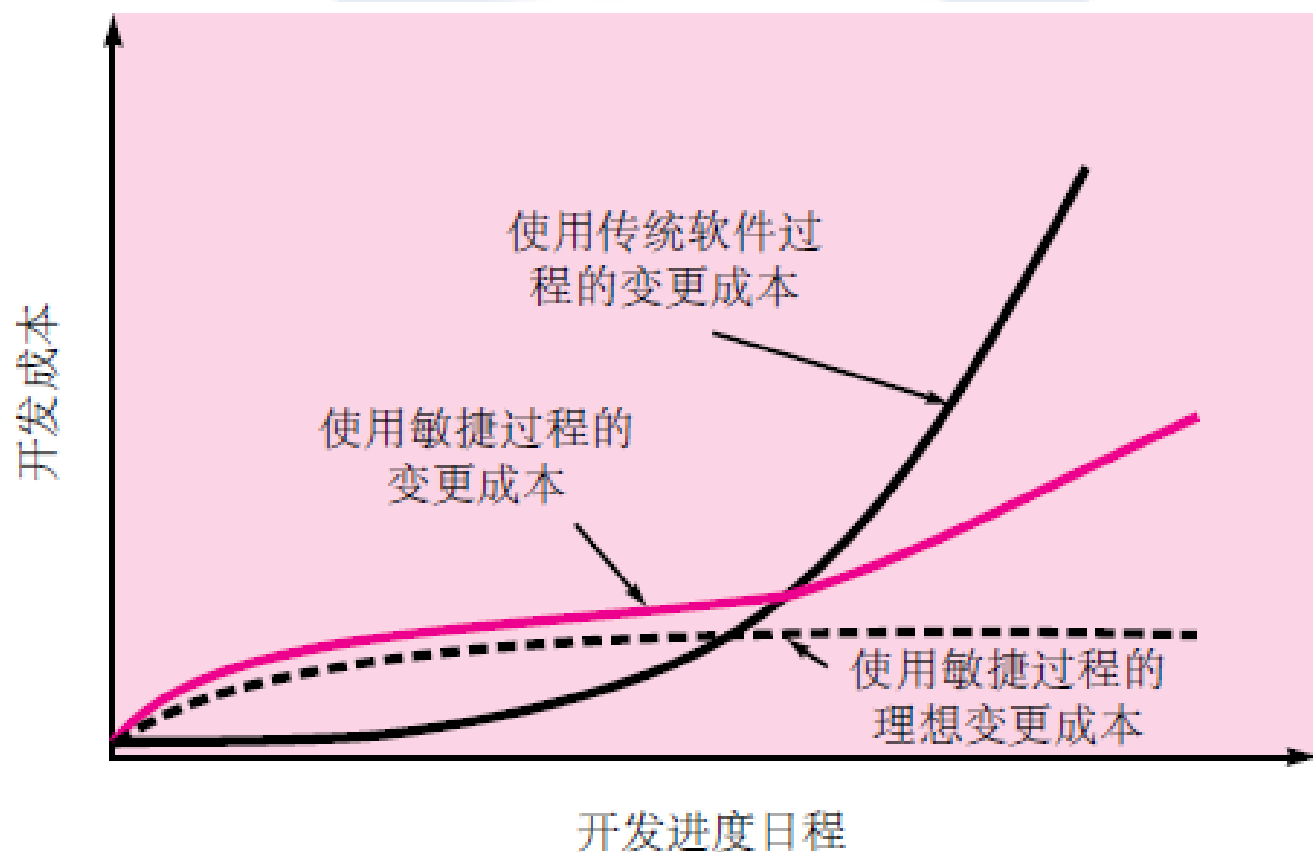
1 什么是敏捷？

- 有效地（快速并适当）响应变更
- 所有利益相关者（经理、客户、最终用户）间的有效沟通
- 客户是开发团队的一部分
- 项目团队高度自治

最重要的是...

- 快速、增量交付可运行软件

2 敏捷及变更成本的费用



3 敏捷过程

敏捷开发也是一个过程，也需要研究和建模。其特征约有：

- 由用户需求的应用场景驱动
- 意识到计划时间很短
- 着重强调构建活动的软件迭代开发
- 交付多个软件增量版本
- 能做出适应性变更

敏捷原则（12条） - I

1. 我们最优先要做的是通过尽早、持续地交付有价值的软件来使客户满意。
2. 即使在开发的后期，也欢迎需求的变更。敏捷过程利用变更为客户创造竞争优势。
3. 经常交付可运行软件，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。
5. 围绕有积极性的个人构建项目。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. 在开发团队内部，最富有效果和效率的信息传递方法是面对面交谈。

敏捷原则（12条） - II

- 7. 可运行软件是进度的首要衡量标准。
- 8. 敏捷过程提倡可持续的开发速度。责任人（**sponsor**）、开发者和用户应该能够长期保持稳定的开发速度。
- 9. 持续关注优秀的技能和好的设计会增强敏捷能力。
- 10. 简单（即：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。

人的因素

- 构造可以满足人员及团队需求的过程模型，而非其他可选的过程模型
- 敏捷开发团队成员及团队本身必须具备以下关键特征：
 - 基本的竞争能力
 - 共同目标
 - 精诚合作
 - 决策能力
 - 模糊问题解决能力
 - 相互信任和尊重
 - 自组织

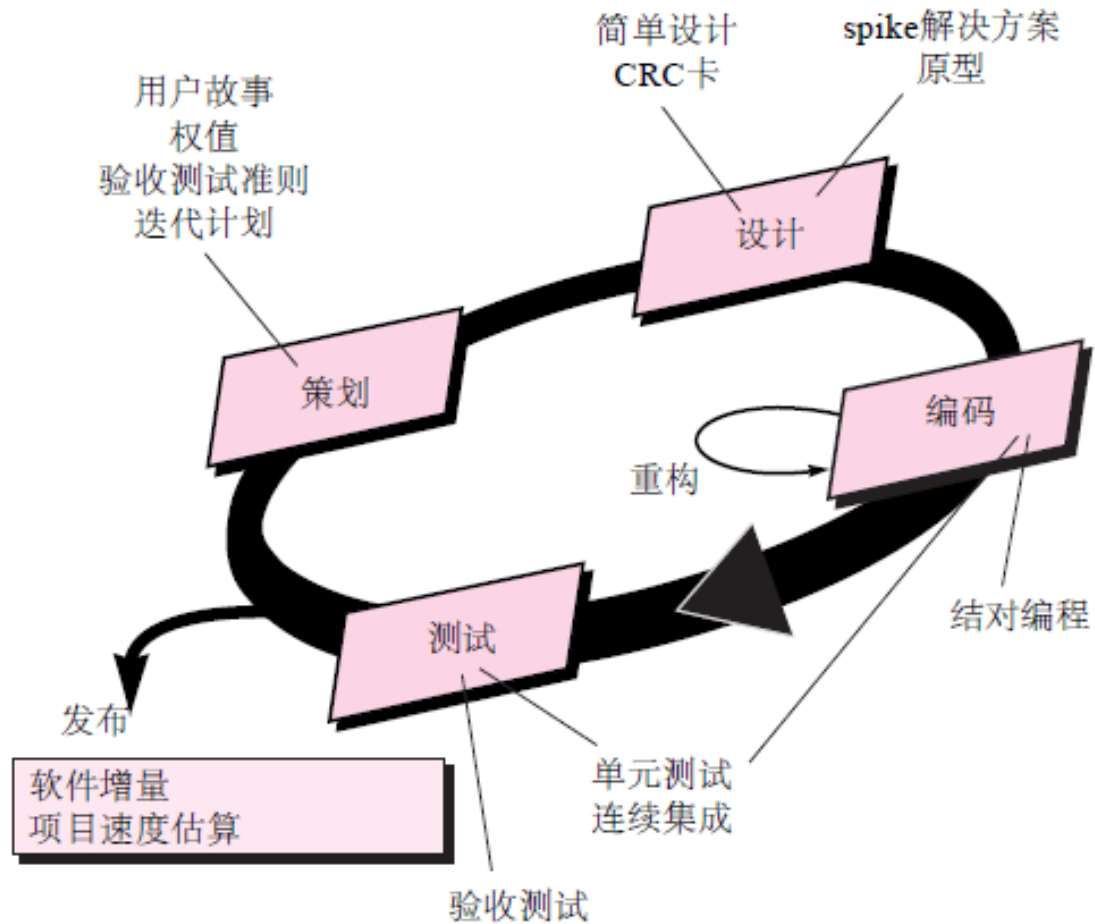
4 极限编程(XP)

- 使用最广泛的敏捷过程，最初由Kent Beck提出。
- XP 策划
 - 以倾听客户的“用户故事”产生开始
 - 敏捷团队评估每个故事和以周为单位的开发成本。过大的故事（如大于3个开发周）要进一步分解。
 - 团队将故事分组到一个可交付增量
 - 承诺交付日期
 - 在第一个软件增量交付之后，项目速度用于帮助确定后续发行版本的交付日期

极限编程(XP)

- XP 设计
 - 遵循KIS (Keep It Simple, 保持简洁) 原则
 - 鼓励使用CRC (Class-Responsibility-Collaborator, 类、职责、协作者) 卡(参见第10章)
 - 对于困难的设计问题, 提出建立 “spike解决方案” —— 一种设计原型
 - 鼓励 “重构” —— 保证外部行为不便的内部程序设计的迭代改进
- XP 编码
 - 推荐在编码开始之前建立单元测试。
 - 鼓励 “结对编程”
- XP 测试
 - 单元测试应当使用一个可以自动实施的框架, 每天测试。
 - 验收测试由客户定义, 并由客户执行评估客户可见的功能特征

极限编程(XP)



工业极限编程 (IXP)

- IXP是XP的一种有机进化，与原来XP的主要差别在于其管理具有更大的包容性，它扩大了用户角色，升级了技术实践。
- IXP合并了六个新实践：
 - 准备评估
 - 项目社区
 - 项目特许
 - 测试驱动管理
 - 回顾
 - 持续学习

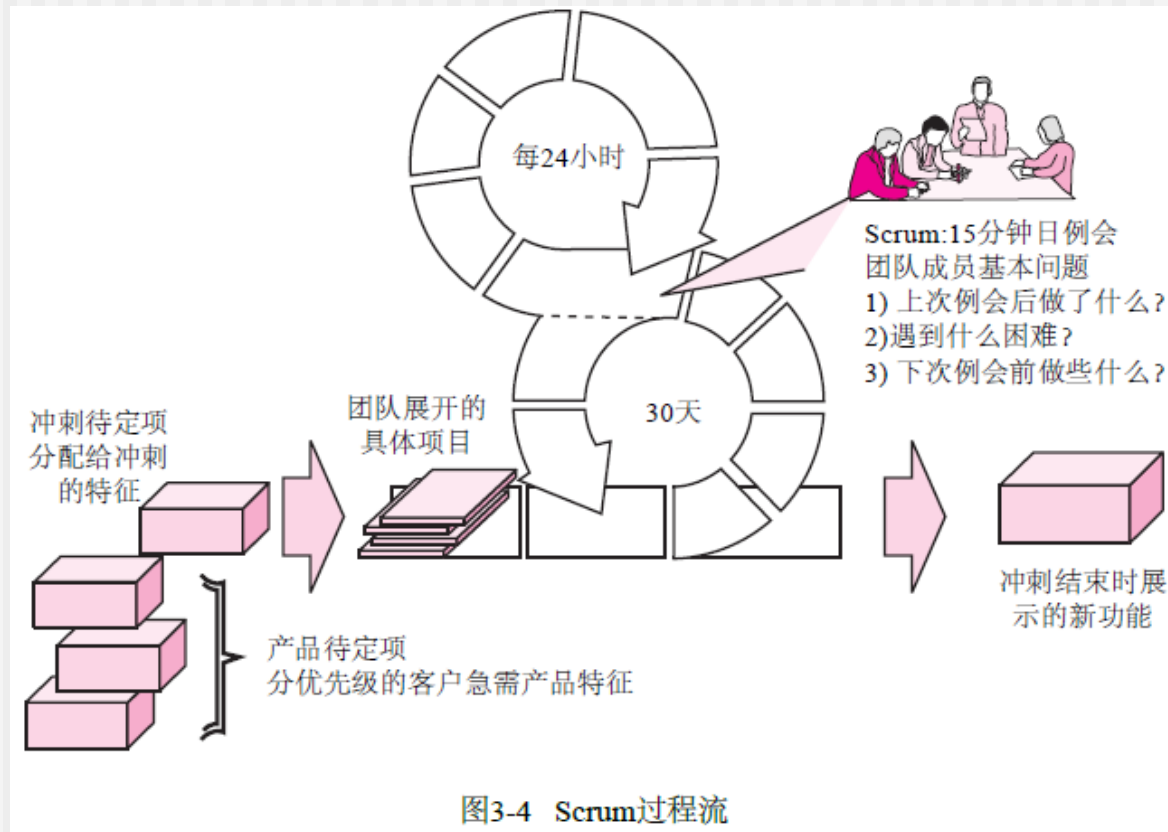
5 其他敏捷过程模型

- Scrum敏捷过程模型
- 动态系统开发方法DSDM
- 敏捷建模AM
- 敏捷统一过程AUP

Scrum

- 由Schwaber and Beedle提出
- Scrum—基本特征
 - 开发活动由工作单元(packets)组成
 - 测试和文档编制工作贯穿始终
 - 发生于一个过程模式中的工作任务称为一个冲刺(sprint)，其来源于待定项(backlog)中定义的需求
 - 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

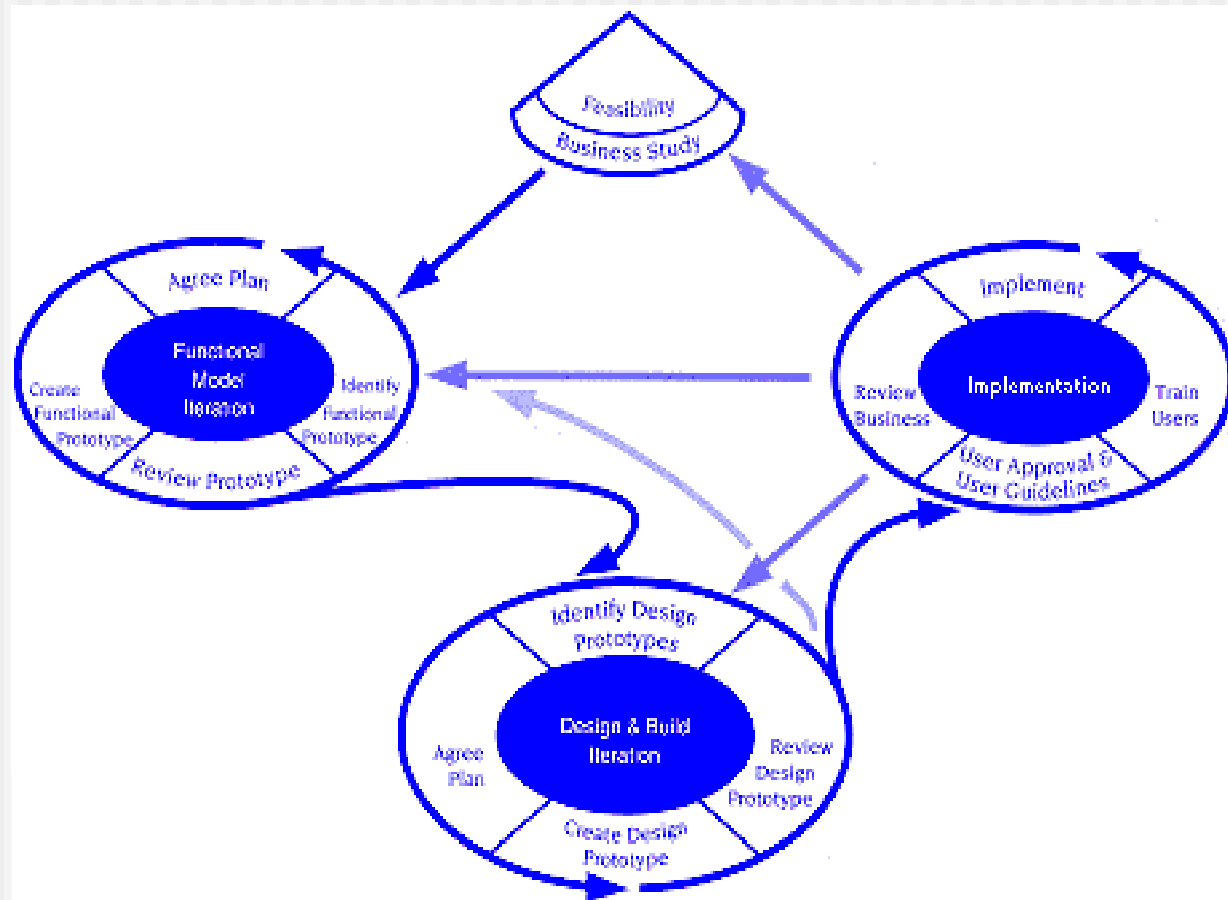
Scrum



动态系统开发方法(DSDM)

- 由DSDM 协会提出(www.dsdm.org)
- DSDM—基本特征
 - 在很多方面类似极限编程
 - 九条基本原则
 - 用户必须持续参与。
 - 必须授予**DSDM**团队制定决策的权力。
 - 注重产品的经常交付。
 - 满足业务用途是接受交付品的主要依据。
 - 迭代和增量式开发对得到正确的业务解决方案是必不可少的。
 - 开发过程中的所有变化可逆。
 - 在高层次上制定需求的基线。
 - 测试自始至终贯穿于开发周期之中。
 - 所有项目涉众间的通力合作是不可或缺的。

动态系统开发方法



DSDM Life Cycle (with permission of the DSDM consortium)

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 8/e (McGraw-Hill, 2014) Slides copyright 2014 by Roger Pressman.

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