哈尔滨工程大学
Harbin Engineering University

软件工程基础 —— 第5章 敏捷开发

计算机学院 孟宇龙

目录

关键概念

- 验收测试
- 敏捷联盟
- 敏捷过程
- 敏捷统一过程
- 敏捷
- 敏捷原则
- 变更成本
- 动态系统开发方法
- 极限编程
- 工业XP
- 结对编程
- 敏捷开发战略
- 项目速度

Kent Beck, *Mike Beedle*, *Arie van Bennekum*, *Alistair Cockburn*, *Ward Cunningham*, *Martin Fowler*, *James Grenning*, *Jim Highsmith*, *Andrew Hunt*, *Ron Jeffries*, *Jon Kern*, *Brian Marick*, *Bob Martin*, *Stephen Mellor*, *Ken Schwaber*, *Jeff Sutherland*, and *Dave Thomas*

敏捷宣言

“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，虽然上述右边的各项很有价值，但我们认为左边的各项具有更大的价值。”

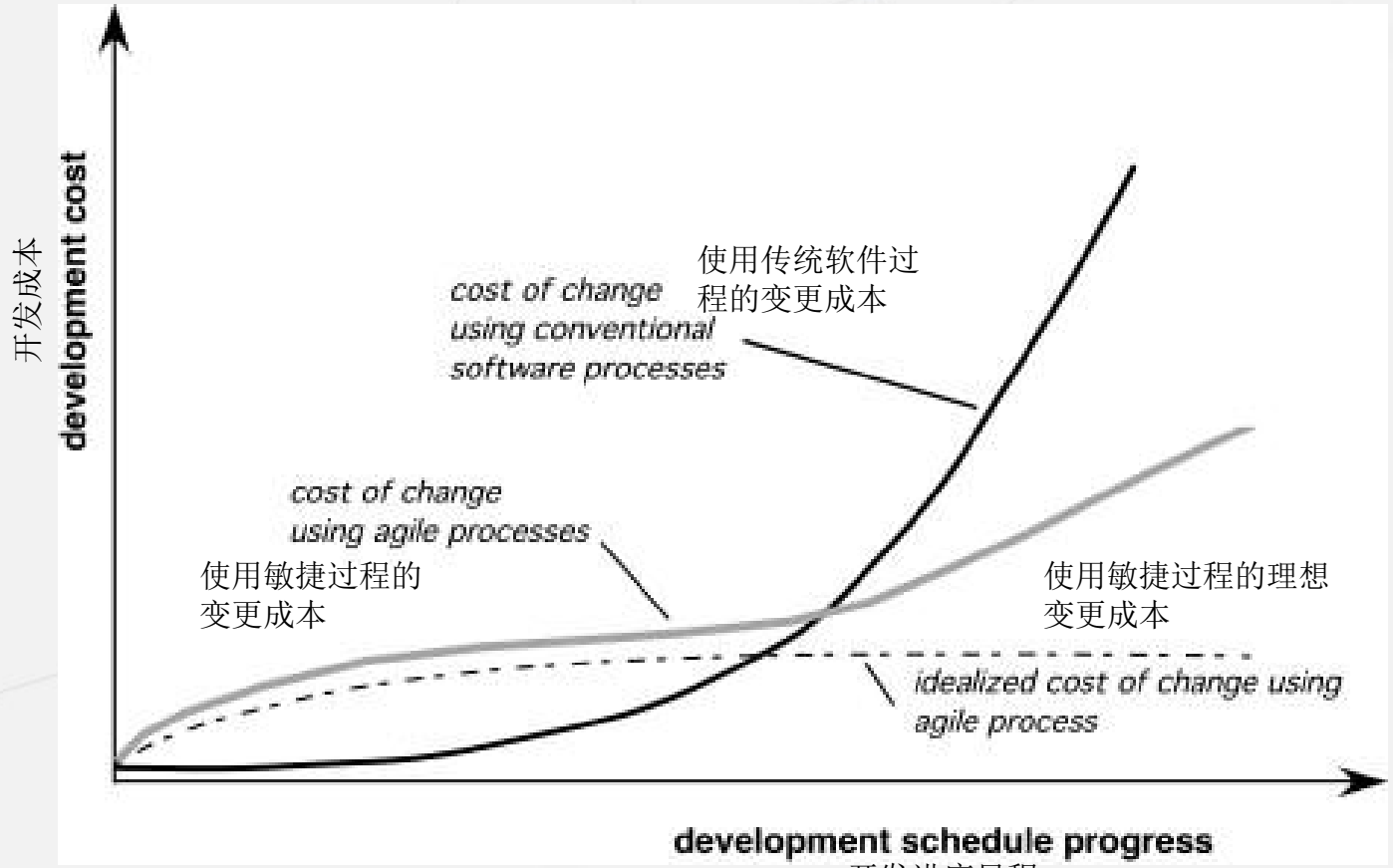
——*Kent Beck et al*

5.1 什么是敏捷

除了响应变更，还有对宣言中提及的思想的信奉

- 有效的（快速并适应）响应变更
- 所有利益相关者之间的有效沟通
- 吸引客户进入团队
- 组织团队使其控制工作的执行
- 产品…
- 快速、增量交付的软件

5.2 敏捷及变更的成本费用



5.3 敏捷过程

以某种方式提出若干假设

- 是由客户对他们需求的描述（场景）所驱动的
- 意识到计划是短期的
- 着重强调构建活动的软件迭代开发
- 交付多个软件增量
- 适应变更的出现

5.3.1 敏捷原则

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。
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。责任人、开发者和用户应该能够长期保持稳定的开发速度
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

- 不必在敏捷和软件工程之间做选择
- 只需要定义敏捷的软件工程方法

5.3 敏捷过程

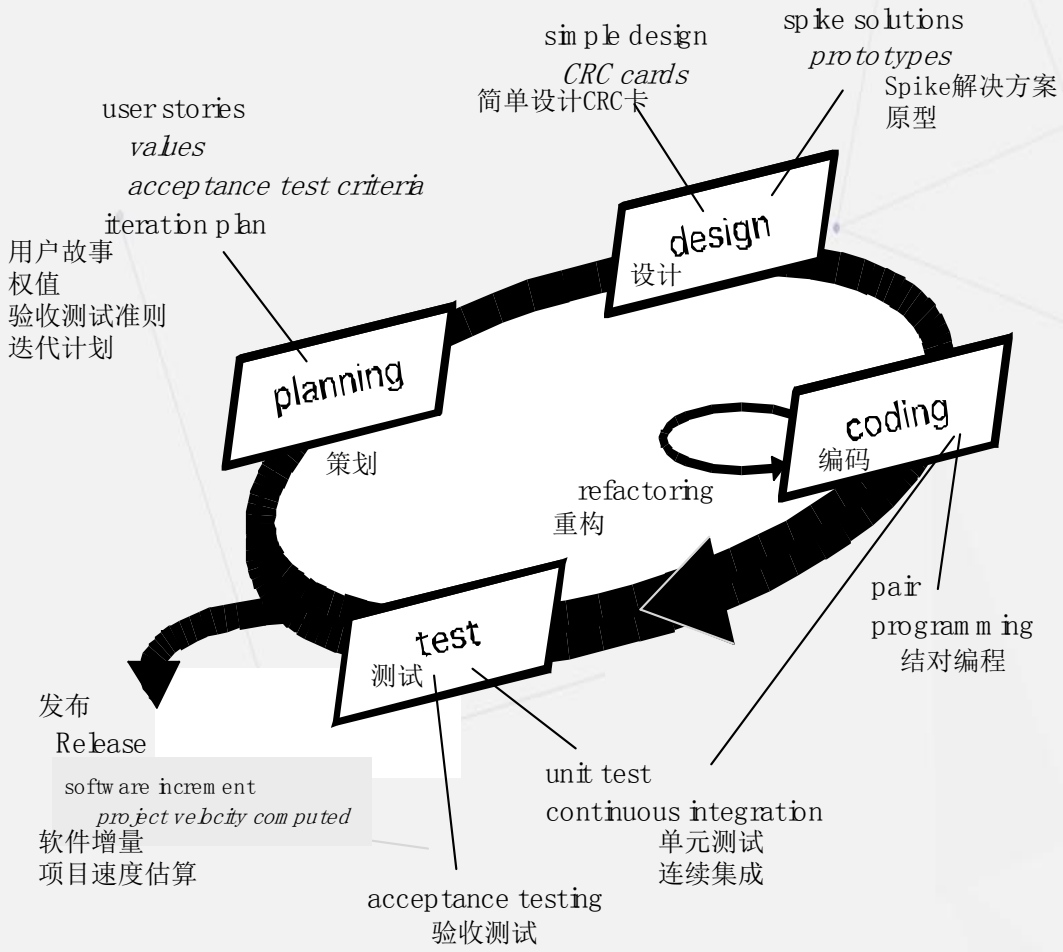
- 满足人员及团队需求的过程模型，而非周围其他过程模型
- 敏捷开发团队成员及团队本身必须具备的关键特征：
  - **基本能力**
  - **共同目标**
  - **精诚合作**
  - **决策能力**
  - **模糊问题解决能力**
  - **相互信任和尊重**
  - **自组织**

极限编程的权值

- 使用最广泛的敏捷过程，最初由Kent Beck提出。

- 五个有重要意义的要素：
  - 沟通
  - 简明
  - 反馈
  - 鼓励
  - 尊重

5.4.1 极限编程过程



极限编程过程

----XP 策划----

- 以“用户故事”的产生开始
- 敏捷团队评估每个故事和分配成本
- 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

----XP 设计----

- 遵循KIS（保持简洁）原则
- 鼓励使用CRC卡(参见第9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”——内部程序设计的迭代改进

----XP 编码----

在编码之前，对于每一个软件增量建议构建一个单元测试

鼓励“结对编程”

----XP 测试----

每天执行所有单元测试

验收测试由客户定义，并由客户执行来评估客户可见的功能特征

5.4.2 工业极限编程

- 能力是你能做什么
- 激励决定了你做什么
- 态度决定了你做的怎么样

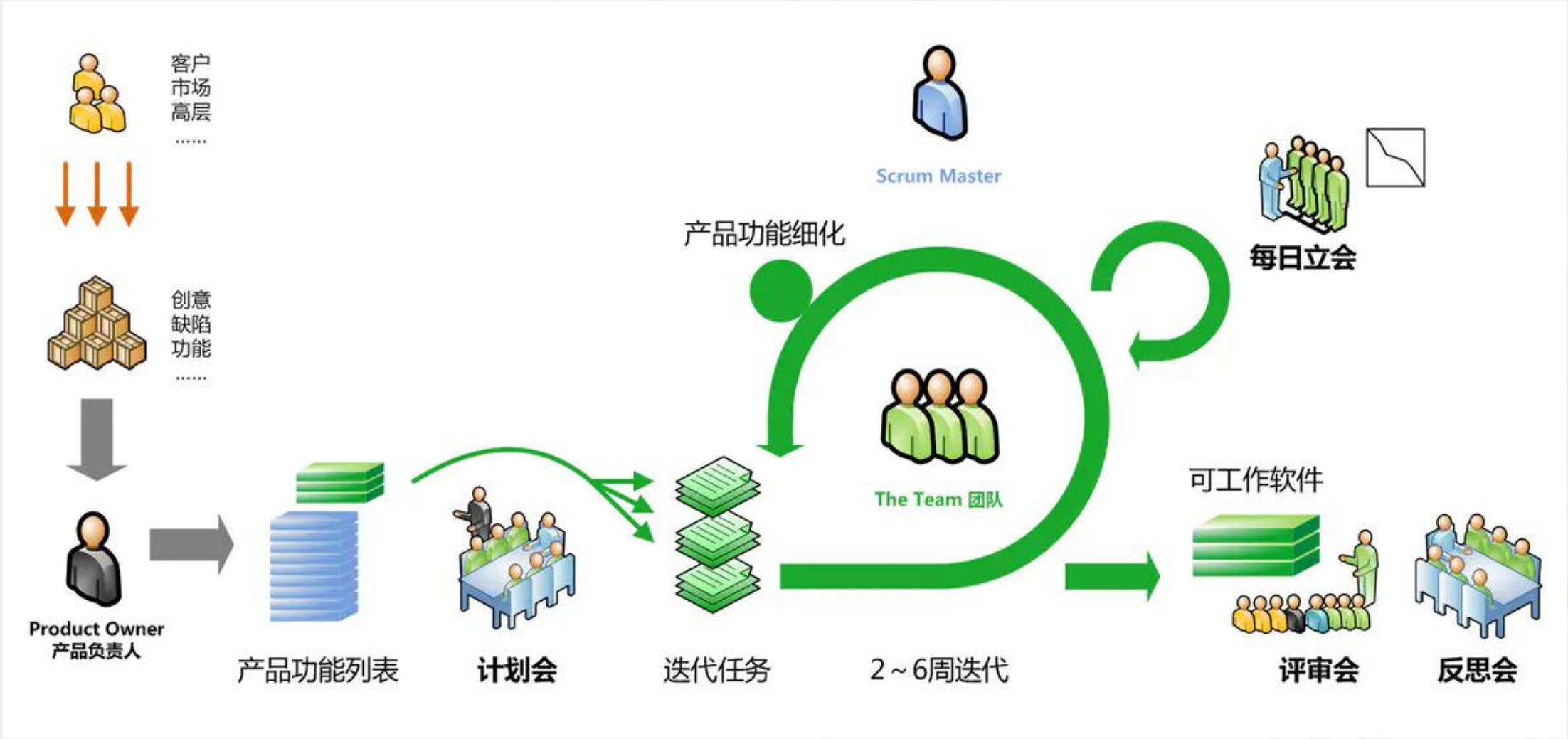
- IXP对管理具有更大的包容性，它扩大了用户角色，升级了技术实践
- IXP 合并了6个新实践：
  - 准备评估
  - 项目社区
  - 项目特许
  - 测试驱动管理
  - 回顾
  - 持续学习

5.5 其它敏捷过程模型

5.5.1 Scrum

- Scrum——区分特征
  - 开发工作划分为“包packet”
  - 测试和文档在产品的构建过程中是持续的
  - 工作任务称为一个“冲刺sprint”，并来自现有需求的一个“待办项backlog”
  - 会议非常短暂，甚至有时开会都不用椅子
  - 在分配的时间段内向客户交付“演示demo”

5.5.1 Scrum

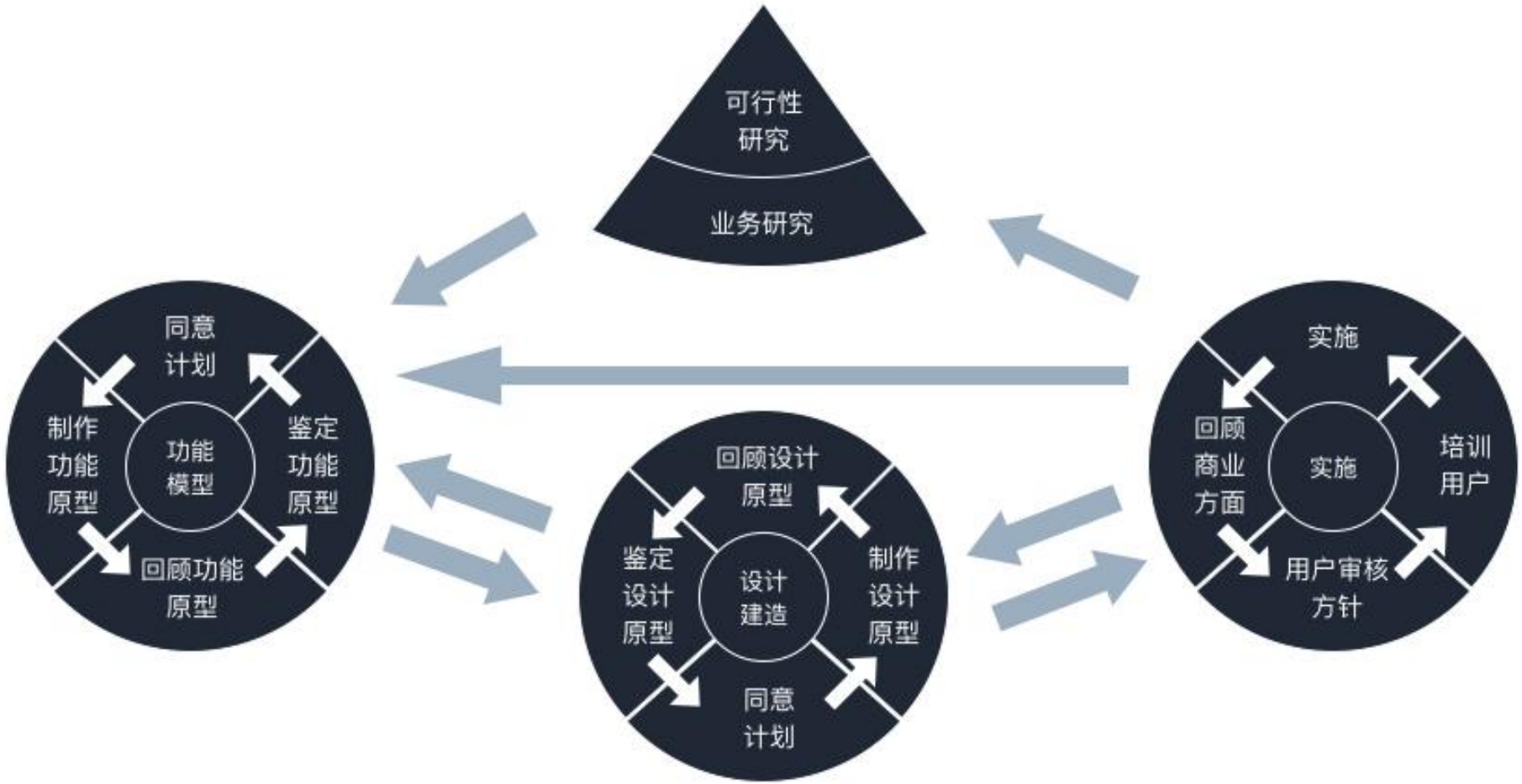


Scrum过程流

5.5.2 动态系统开发方法

- 由 DSDM 协会提出 (www.dsdm.org)
- DSDM——区分特征
  - 在很多方面类似于XP
  - 九条指导原则
    - 活跃的用户参与是必不可少的。
    - DSDM 团队必须有决定权。
    - 注重多次的产品交付。
    - 适应商业目的是交付产品验收的基本标准。
    - 迭代和增量开发有必要聚集于一个精确的业务解决方案。
    - 开发期间所有的变更都是可逆的。
    - 需求基线是高水准的。
    - 测试集中在整个生命周期。

5.5.2 动态系统开发方法



DSDM生命周期（DSDM协会许可）

5.5.3 敏捷建模

最初由 Scott Ambler提出：

- **AM是一种基于实践的方法学，用于对基于软件的系统实施有效建模和文档编制**
- **在软件开发项目中，AM是可以有效并以轻量级方式用于软件建模的标准、原则和实践**
- **由于敏捷模型只是大体完善，而不需要完美，因此敏捷模型比传统的模型更有效**

5.5.3 敏捷建模

- 提出一系列敏捷建模原则
  - 有目的的模型
  - 使用多个模型
  - 轻装上阵
  - 内容重于表现形式
  - 理解模型及用于构建模型的工具
  - 适应本地需要

5.5.4 敏捷统一过程(AUP)

- 采用经典UP阶段性活动
- 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