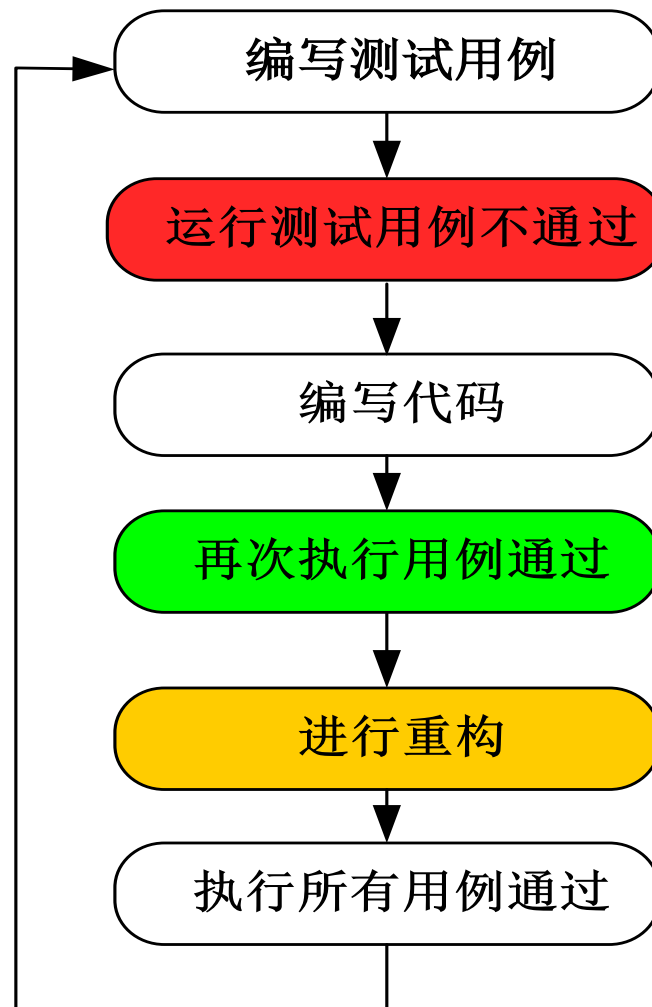
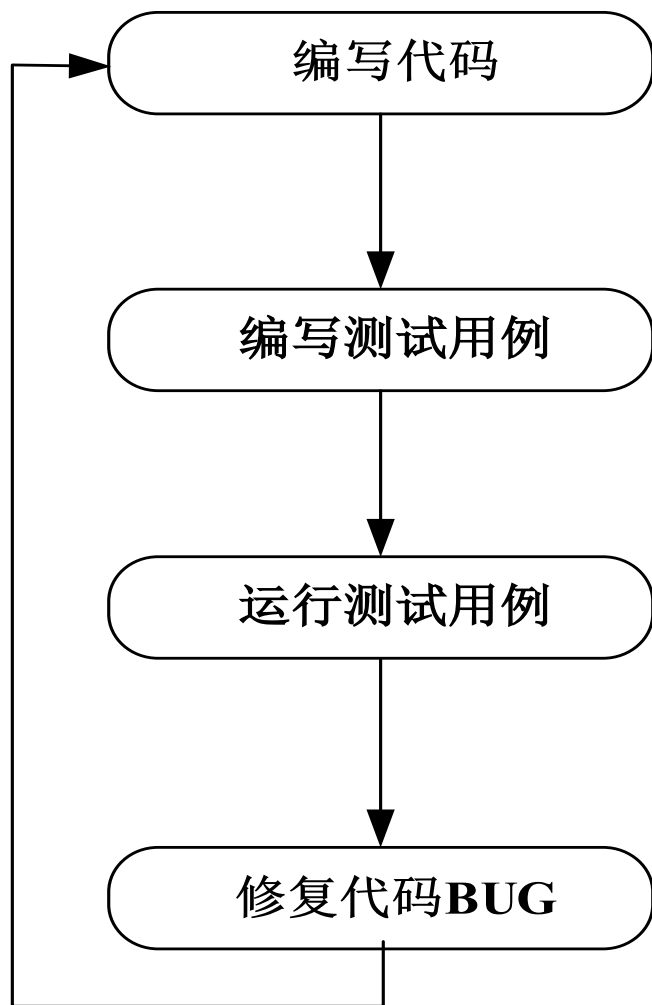


XP的测试

- 测试是XP方法的重要特点
- XP 测试特性：
 - 测试驱动开发 (Test-driven development , TDD)
 - 用户参与测试开发和确认的过程
 - 使用自动化测试框架

TDD开发流程



红灯↓绿灯↓重构

TDD的关键要点

- 测试代码和源代码一样都需要简洁，可读性好；
- 测试用例的设计要保证完备，覆盖被测单元的所有功能；
- 当功能单元较大时，为降低难度，可分解为多个更小的功能单元，并逐一用 TDD 实现。



用户参与

- 用户进行验收测试
 - 定义验收测试用例
 - 验收测试用例来源于用户故事
- “开药” 故事
 - 药物的剂量改变了
 - 选择了新药
 - 在处方集中查找药物
 -



测试自动化

- 开发人员编写测试代码、设计测试用例
- JUnit
- “剂量核对”任务测试用例

测试3：剂量核对

输入：

1. 药物的单次剂量，以 mg 为单位
2. 每天用药的次数

测试：

1. 单次剂量正确，但每天用药次数太多。
2. 单次剂量太多和太少。
3. 单次剂量 \times 用药次数太多和太少
4. 单次剂量 \times 用药次数在允许的范围内

输出：

OK或者错误信息提示剂量超出了安全范围

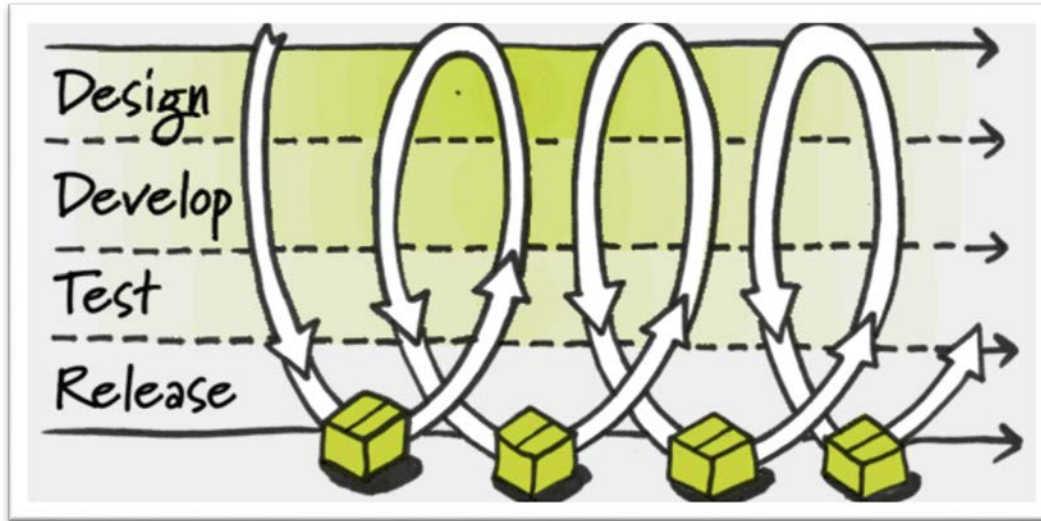


结对编程（Pair programming）



- Erich Gamma与Kent Beck
- 并排坐于计算机旁协作开发程序
 - **驾驶员**：编写代码
 - **领航员**：评审代码并提供反馈
 - 两个程序员定期交换角色
- 好处
 - 新手当驾驶员，经验丰富的程序员当领航员，是一个有效的经验学习过程
 - 开发时让另一双眼睛评审代码有助于在早期发现错误

持续集成(Continuous Integration)



一种软件开发实践，即团队的成员经常集成他们的工作，通常每个成员每天至少集成一次，每次集成通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽快地检测出集成错误。

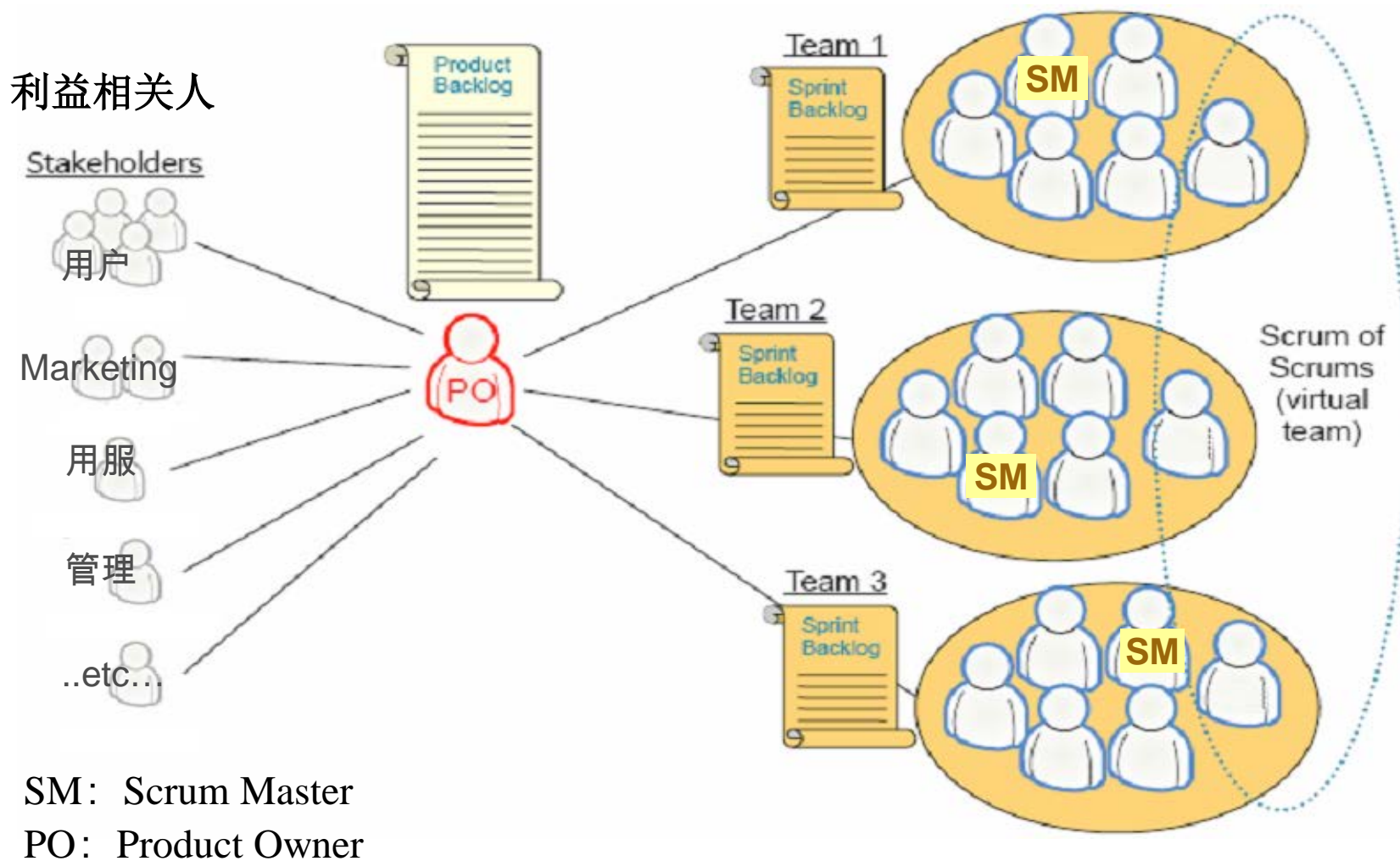
——Martin Fowler

“paper: Continuous Integration”

CI的关键要点

- 持续集成强调 “快速” 和 “反馈”，要求完成一次系统集成的时间尽量短，并提供完备且有效的反馈信息。
- 修复失败的构建是团队最高优先级的任务。
- 开发人员须先在本地构建成功，才可提交代码到版本控制库。
- 开发人员每天至少向版本控制库提交一次代码。
- 每次构建都必须100%通过测试。
- 持续集成的状态必须实时可视化显示给所有人。

SCRUM敏捷团队



敏捷团队的三个核心角色

■ PO(Product Owner)

- **确保Team正确的事**
- 代表利益相关人，对产品投资回报负责
- 定义产品需求并确定优先级
- 重构、持续集成环境搭建等内部任务

■ Scrum Master(Scrum教练)

- **确保Team正确地做事**
- 辅导团队正确应用敏捷实践
- 引导团队建立并遵守规则
- 激励团队

敏捷团队的三个核心角色

■ Team(开发团队)

- 负责产品需求实现
- 负责估计工作量
- 根据自身能力找出最佳方案去完成任务
- 保证交付质量
- 向PO和利益相关人演示工作成果
- 团队自我管理、持续改进

敏捷管理实践：每日站立会议



- 每日工作前，团队成员的例行沟通机制，由Scrum Master组织，Team成员全体**站立**参加
- 聚焦在下面的三个主题：
 - 我昨天为本项目做了什么？
 - 我计划今天为本项目做什么？
 - 我需要什么帮助以更高效的工作？

关键点

- 准时开始：按计划会议制定的时间地点开会，形成团队成员的自然习惯；
- 高效会议：会议限时**15分钟**，每个人都保持站立，依次发言，不讨论与会议三个主题无关的事情（如技术解决方案等）；
- 问题跟踪：Scrum Master应该记录下所有的问题并跟踪解决。

总结

- 过程模型的选择可根据用户、客户和开发人员的需要进行定义和裁剪
- 实际项目中，应该用一组过程模型描述软件开发过程，而不是集中于单个模型
- 不论使用哪一种过程模型，许多活动都是所有模型共有的
 - 基本活动
 - 普适性活动