

百味博客

敏捷概述

内容来源于互联网，百味博客收集整理，如有侵权,请及时联系

敏捷简史

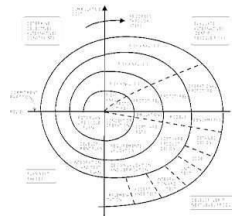
CMM



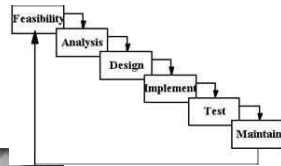
ISO



重量级方法



增量、螺旋模型



原型模型



瀑布模型

软件危机



敏捷方法

1960

1970

1980

1990

2000

2010

敏捷宣言



我们正在通过亲身实践以及帮助他人实践，揭示更好的软件开发方法。通过这项工作，我们认为：

个体和交互	胜过	过程和工具
可以工作的软件	胜过	面面俱到的文档
客户合作	胜过	合同谈判
响应变化	胜过	遵循计划

虽然右项也具有价值，
但我们认为左项具有更大的价值。

敏捷是在动荡的业务环境中获得利益并创造变化和响应变化的能力

业界敏捷浪潮

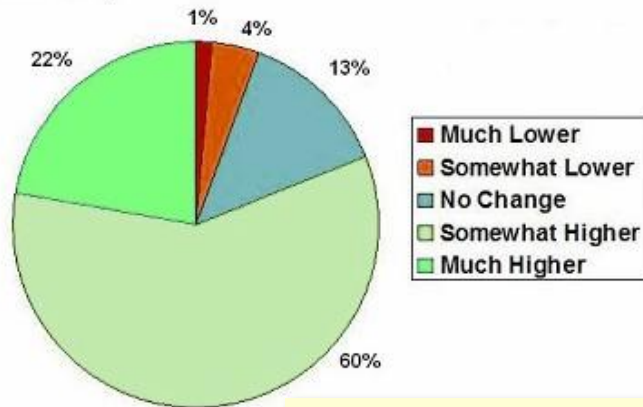


- ISO 9000（09版）标准将在原来八大原则的基础上新增敏捷原则
- 2000年美国军方软件开发标准（DOD 5000.2）推荐迭代为软件开发优选模式
- 世界影响最大的美国波多里奇国家质量奖将敏捷作为核心的十一大原则之一

敏捷对生产率、质量、满意度、成本有明显改进

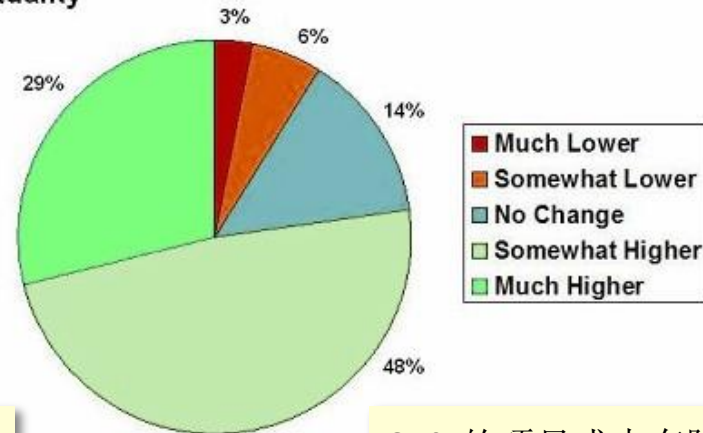
82%的项目生产率有提高

Productivity



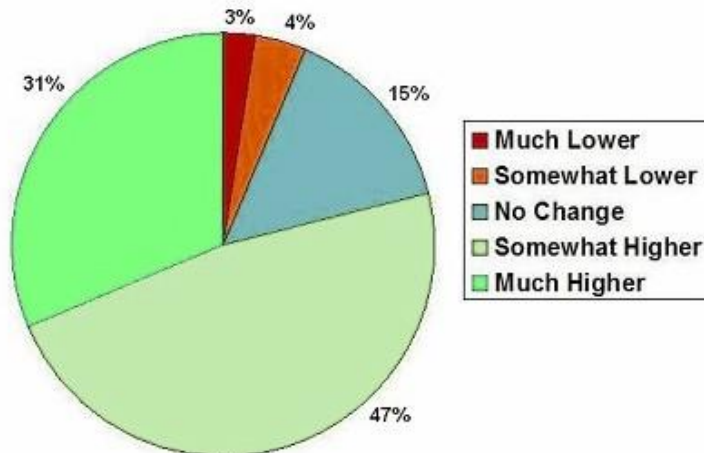
77%的项目质量有提高

Quality



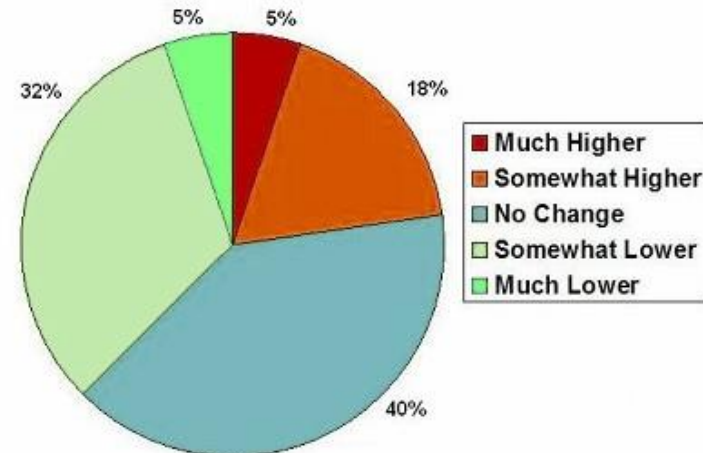
78%的项目客户满意度有提高

Business Stakeholder Satisfaction



37%的项目成本有降低

Cost of System Development

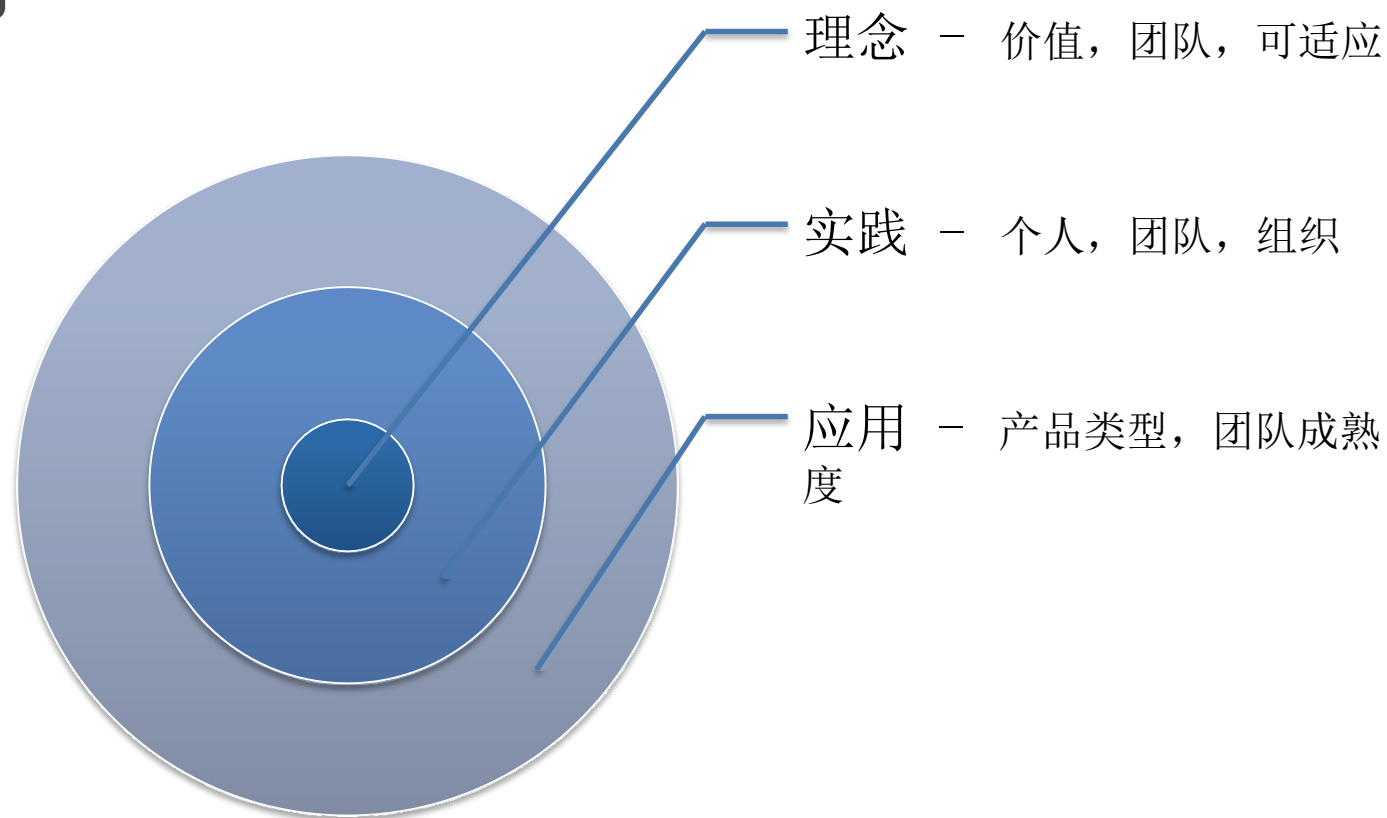


Source: Dr. Dobb's Journal 2008 Agile Adoption Survey

Copyright 2008 Scott W. Ambler

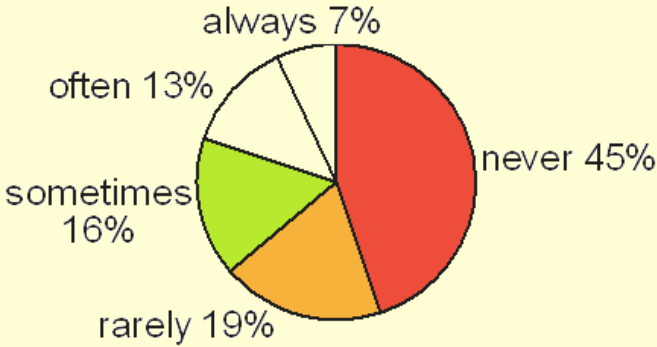
* 以上数据来自DDJ 2008由Scott Ambler发起的网上调查结果

敏捷=理念+优秀实践+具体应用



理念：聚焦客户价值(Value)，消除浪费

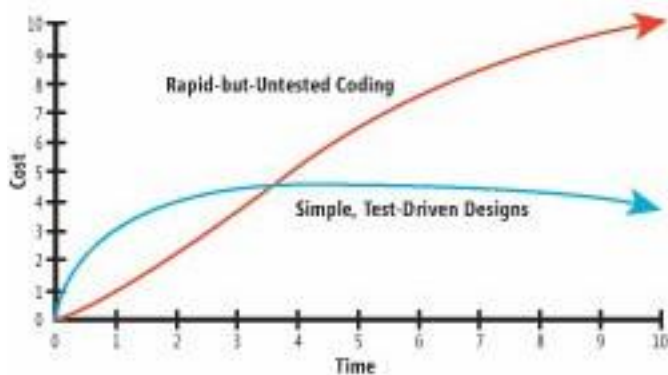
软件业：45%的软件特性客户没有使用



Source: Standish Group 来自5万个软件开发项目的调查

	浪费类别	浪费举例
1	部分完成的工作	部分完成但没有最终落地的工作
2	未应用特性	开发完成但没有被客户应用的特性。
3	再次学习	人员频繁流动导致经验不能积累，反复重新学习；在多个环节移交时，接收信息者也需要重新学习；拥有某领域的专家，但在开发过程中需要此领域经验时，他却没参与，而是团队重新摸索。
4	移交	知识信息的传递总是伴随信息丢失，隐形知识尤其困难，分工过细往往导致过多不必要的移交
5	任务切换	研究表明多任务工作会导致效率下降20%-40%
6	延迟	因任务或资源相互依赖而导致工作停滞
7	缺陷	解决缺陷活动本身就是浪费，而且缺陷越遗留到后端浪费越大。

聚焦客户价值



更改成本曲线

- 进行系统思考

- 全局优化: 用今天的付出换回未来可预见的持续发展。

- 质量内建:

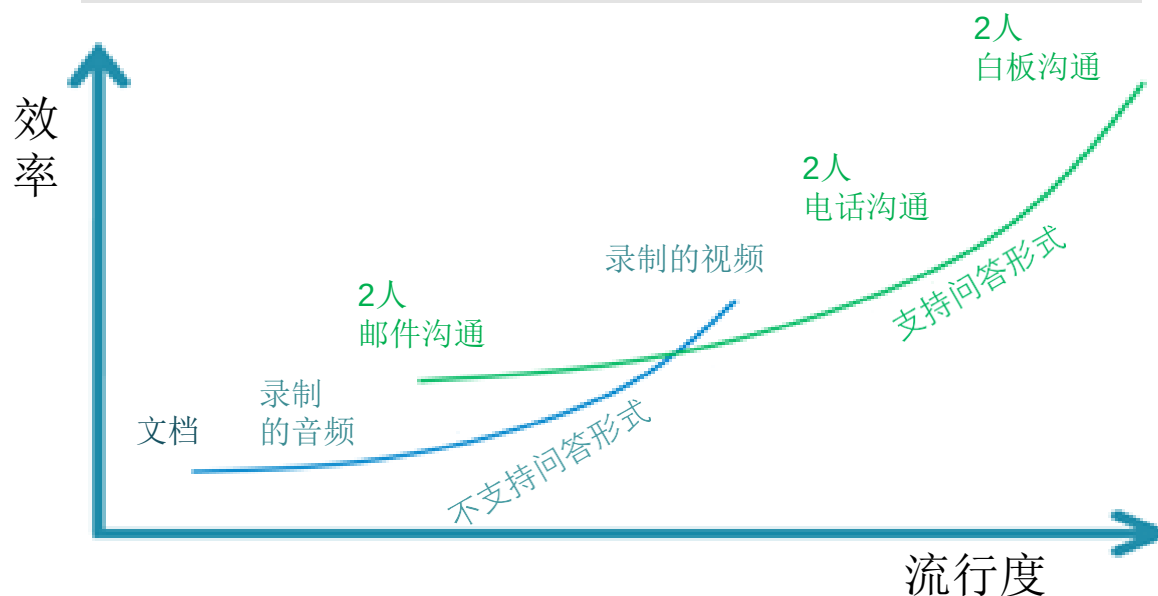
- 形成零缺陷文化，不要容忍缺陷：发现缺陷应立即停下来解决，以保证在坚实的质量基础上前行。
- 开发和测试紧密协作：测试人员参与到设计和开发过程中，共同预防缺陷的产生。

- 严控技术债务

- 找出债务的位置。
- 构建业务案例，并对债务影响领域的优先级达成共识（包括团队和利益相关者）。
- 采用行之有效的策略修复选择处理的债务问题。
- 重复执行上述步骤。

理念：激发团队(Team)潜能，加强协作

研究表明面对面的沟通最有效
业界调查：一个50人开发团队，每人平均30%时间用于编码，
70%的时间用于与其他成员交流。



- * 团队是价值的真正创造者，应加强团队协作、激发团队潜能
- * 软件开发是一种团队活动，首先应做到提升沟通效率降低交流成本

激发团队，敏捷方式下管理者的转变

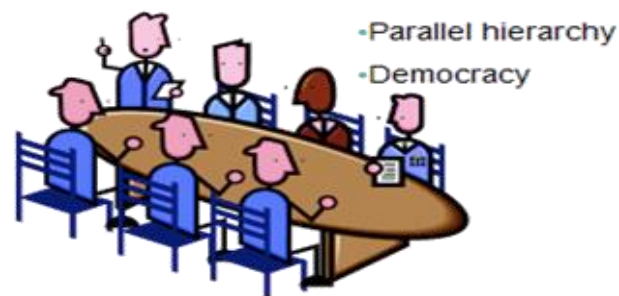
Traditional Organization



管理者努力“控制”团队：

- 制定详细的工作计划，并做出详细的工作安排
- 指令性工作方式
- 监控过程
- 基于复杂规则的管理

Agile Organization



管理者努力“激发”团队：

- 通过目标来牵引团队自主工作
- 帮助团队提供资源，排除障碍
- 营造团队自我管理的工作氛围
- 作为教练辅导团队进步
- 基于简单原则的管理，原则简单但必须被遵守

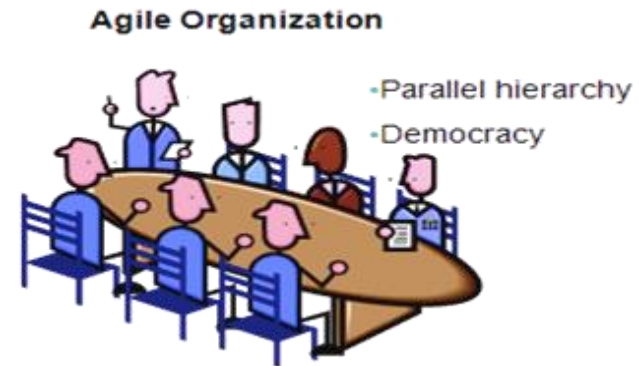
激发团队，敏捷方式下团队成员的转变



团队成员是“听从安排的独立贡献者”：

- 被动等待主管下指令安排工作
- 独立工作为主，协作少
- 以文档和邮件为主要沟通方式
- 只关注个体任务“做完”，不关注团队目标
- 能力相对单一，学习动力不足

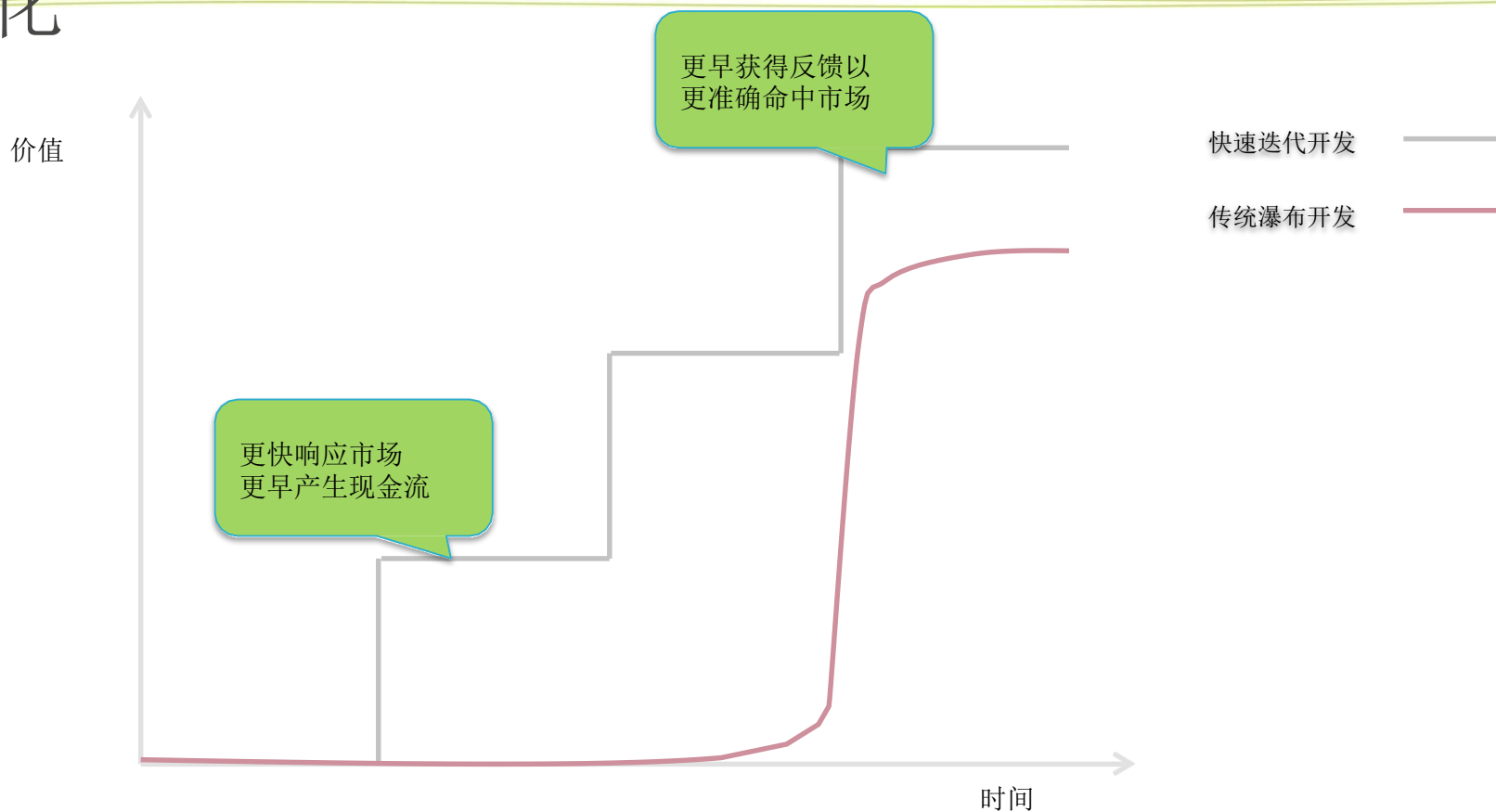
敏捷方式



团队成员是“全方位的积极参与者”：

- 共同参与计划制定和任务安排
- 团队协作贯穿工作始终
- 面对面交流是主要沟通方式
- 关注团队目标，共担责任
- 能力要求更广，主动学习适应岗位要求

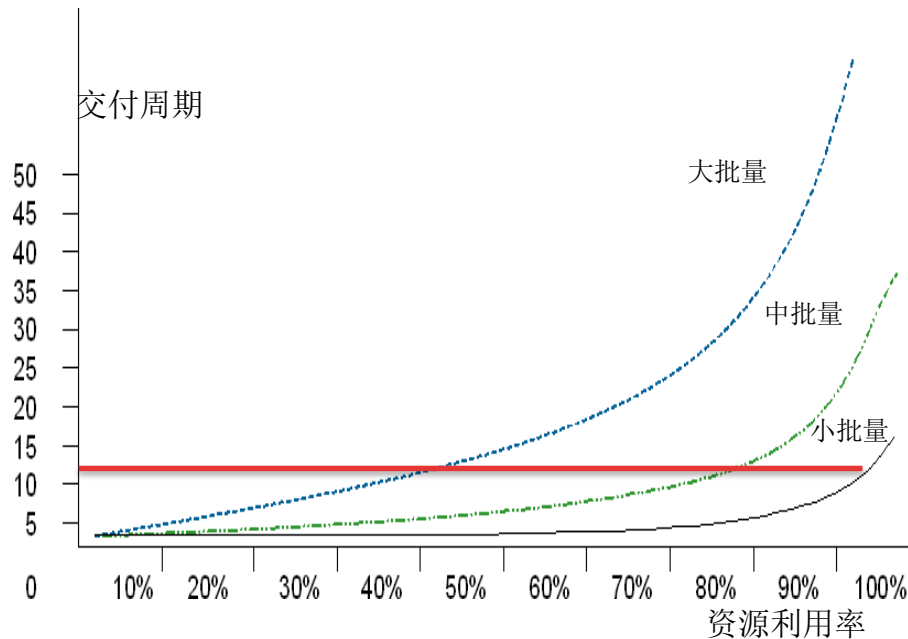
理念：不断调整以适应(Adapting)变化



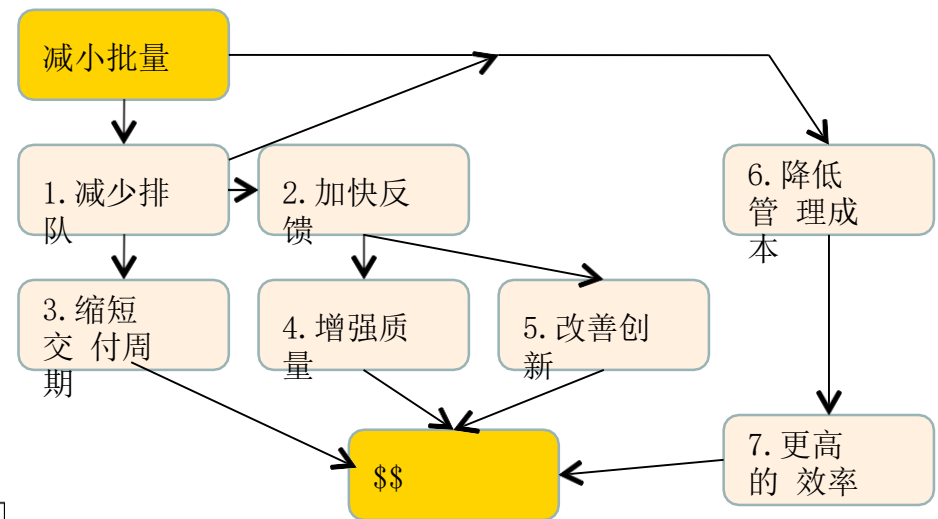
- 《人月神话》：软件开发是人类最复杂工作之一，软件具有四个属性：复杂性、一致性、可变性和不可见性。
- 软件开发是不可重复、探索性的、演进的，适应性过程。
- 不断的根据经验调整，最终交付达到业务目标的产品。

适应变化，小批量是快速交付的关键

排队理论：小批量与缩短交付周期、人员有效产出的关系



减少批量的好处

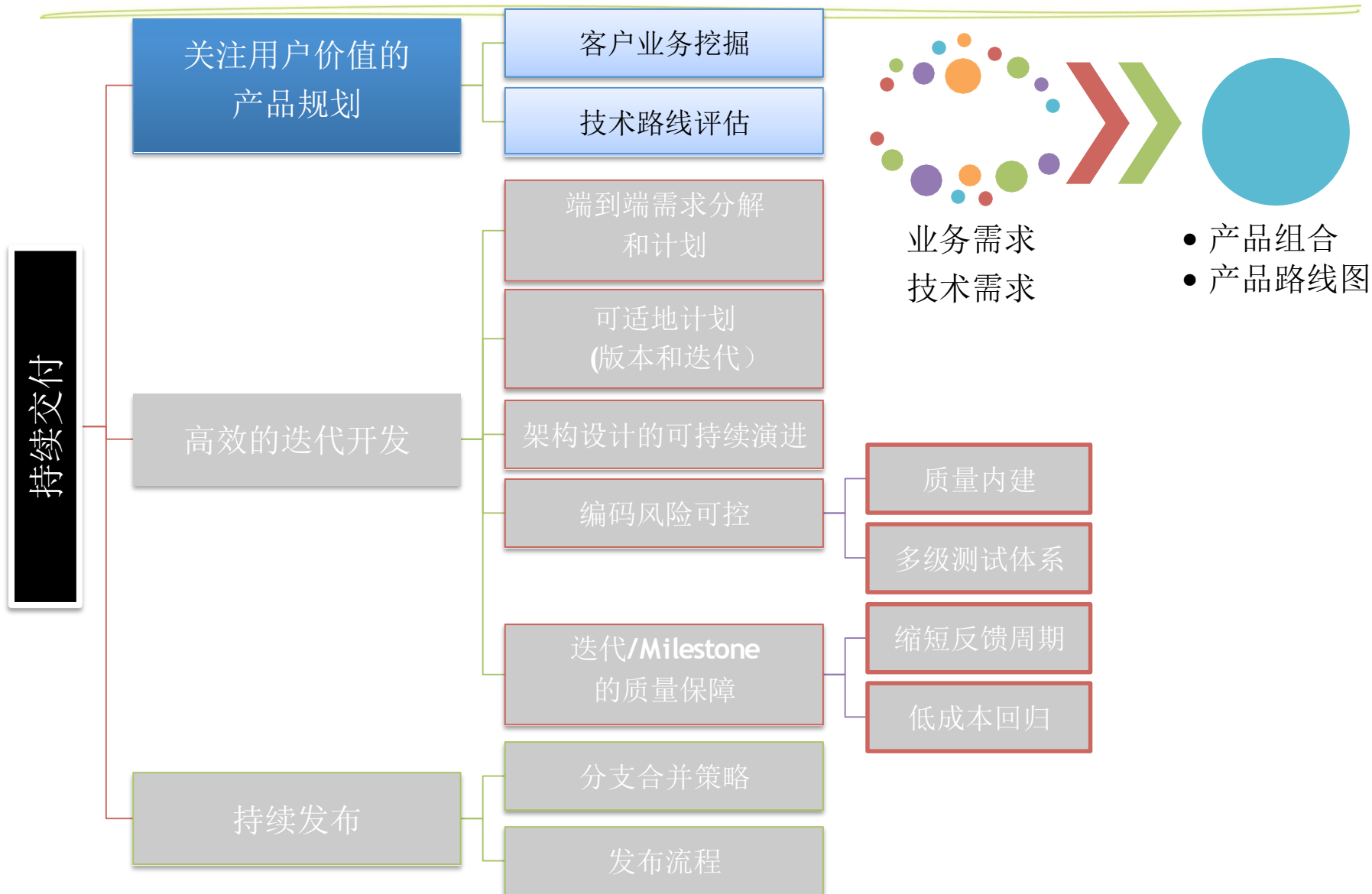


Source: Craig Larman

- 通过尽早地、持续地交付有价值的软件来使客户满意。
 - 经常性的交付可以工作的软件，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。
- 摘自敏捷软件的十二个原则

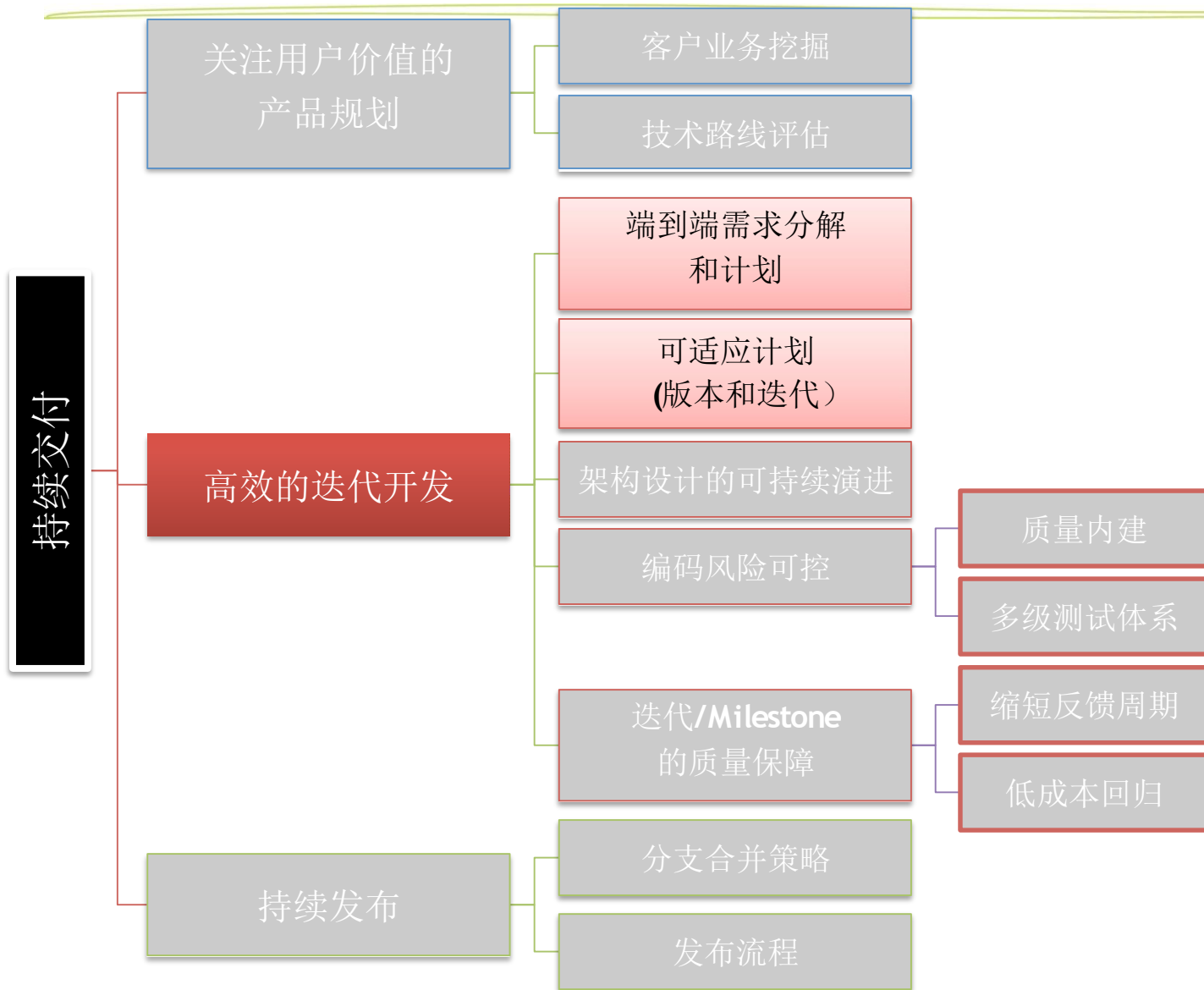
实践：以持续交付为目标的开发活动组织

关注价值的创造，而非任务的完成



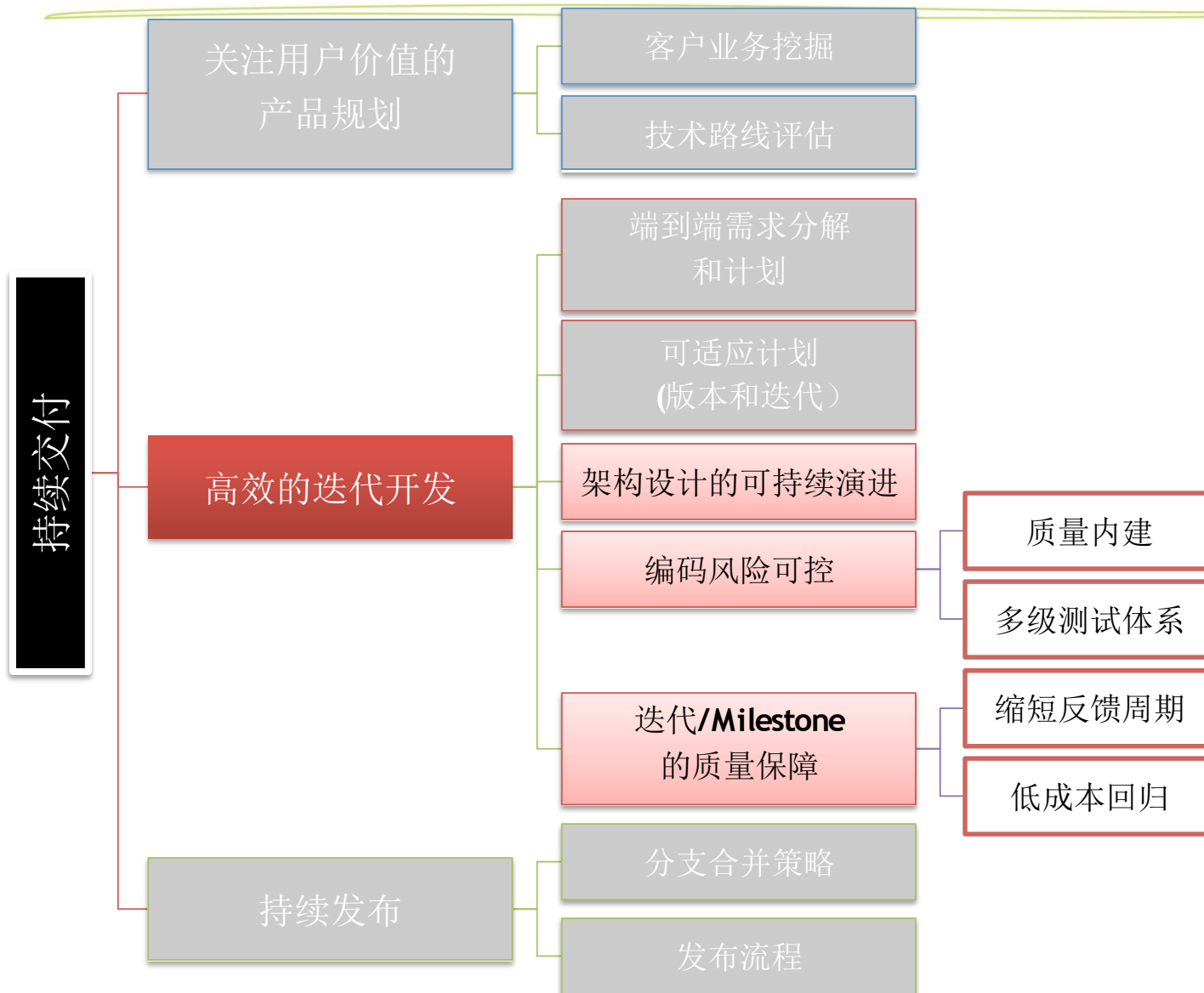
实践：以持续交付为目标的开发活动组织

关注价值的创造，而非任务的完成



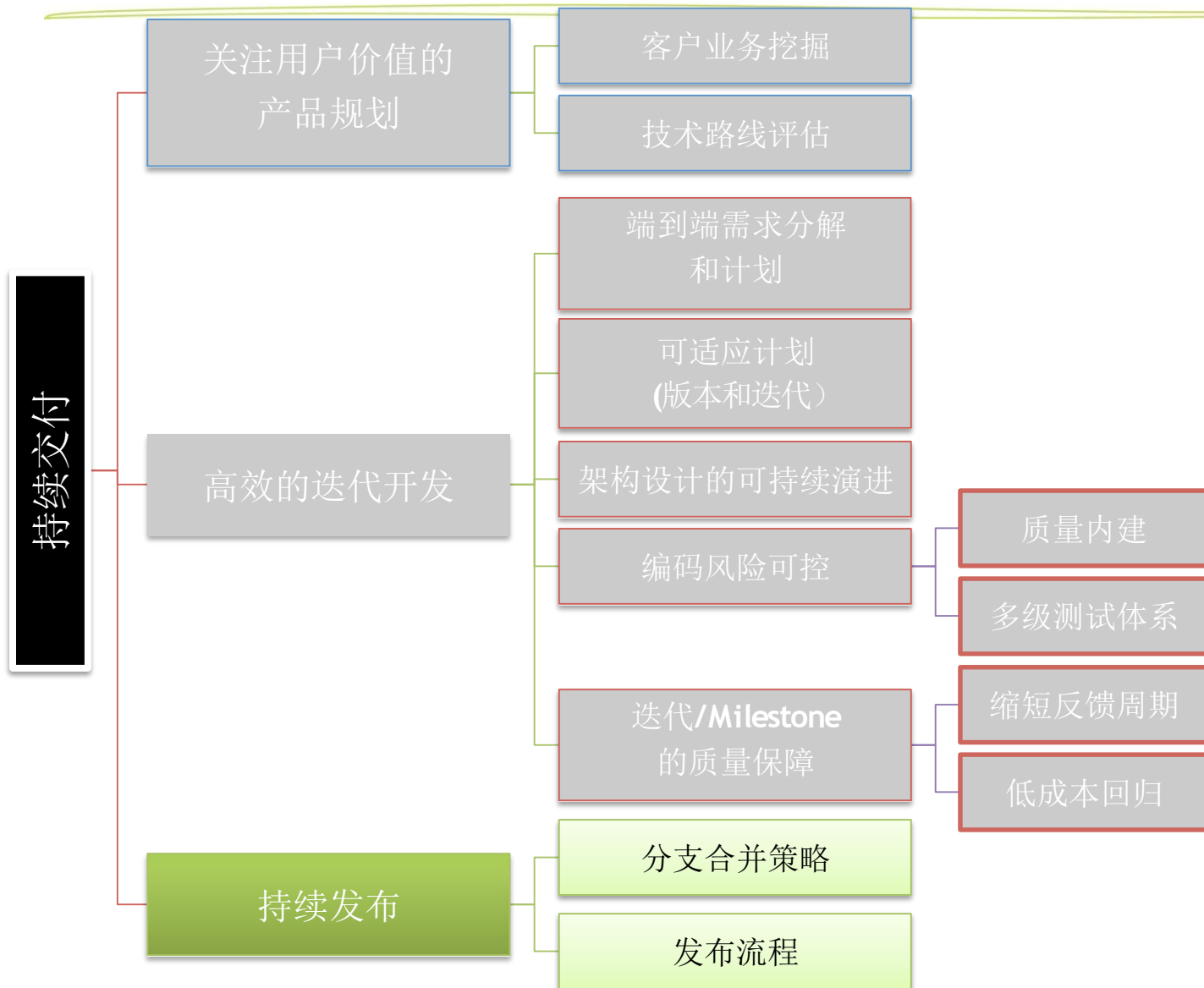
实践：以持续交付为目标的发展活动组织

关注价值的创造，而非任务的完成

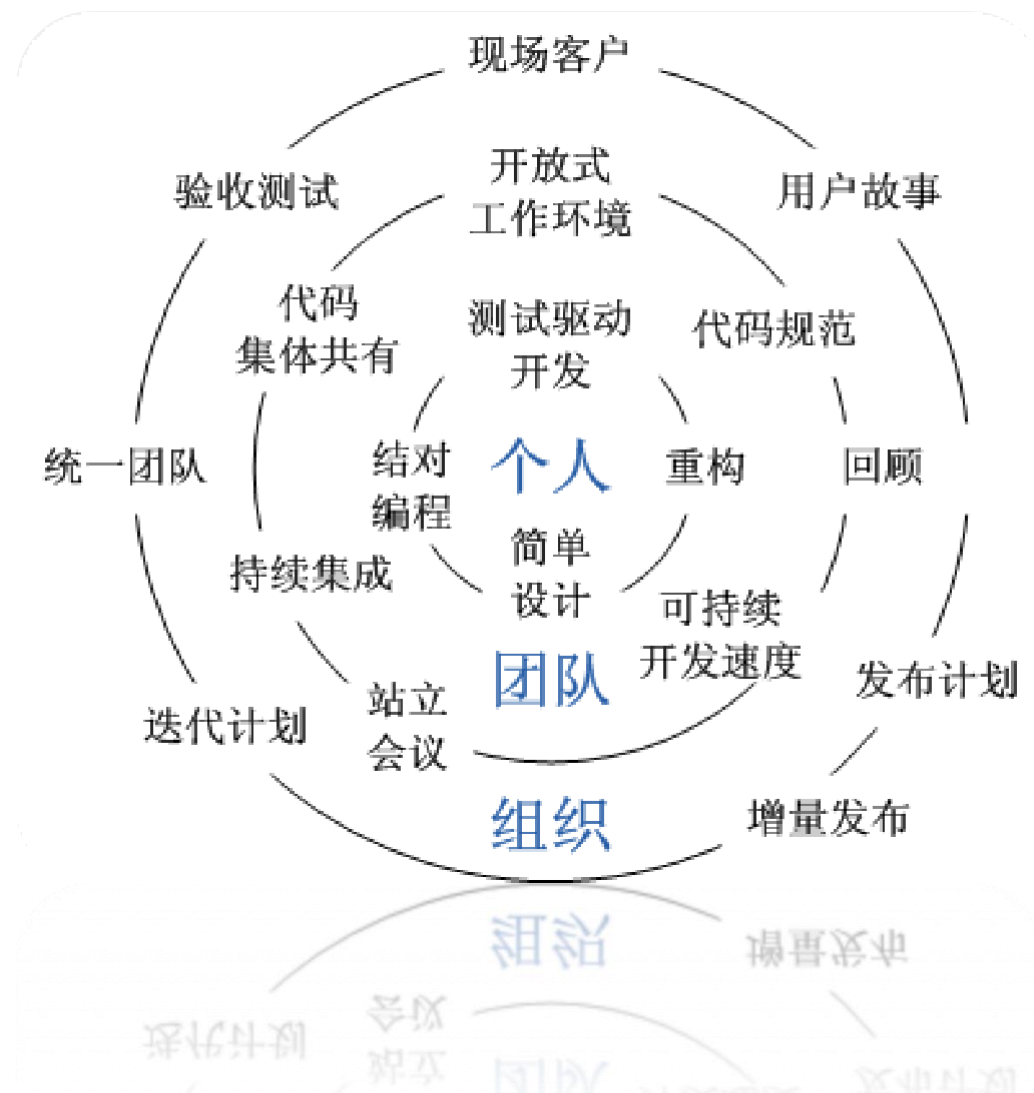


实践：以持续交付为目标的发展活动组织

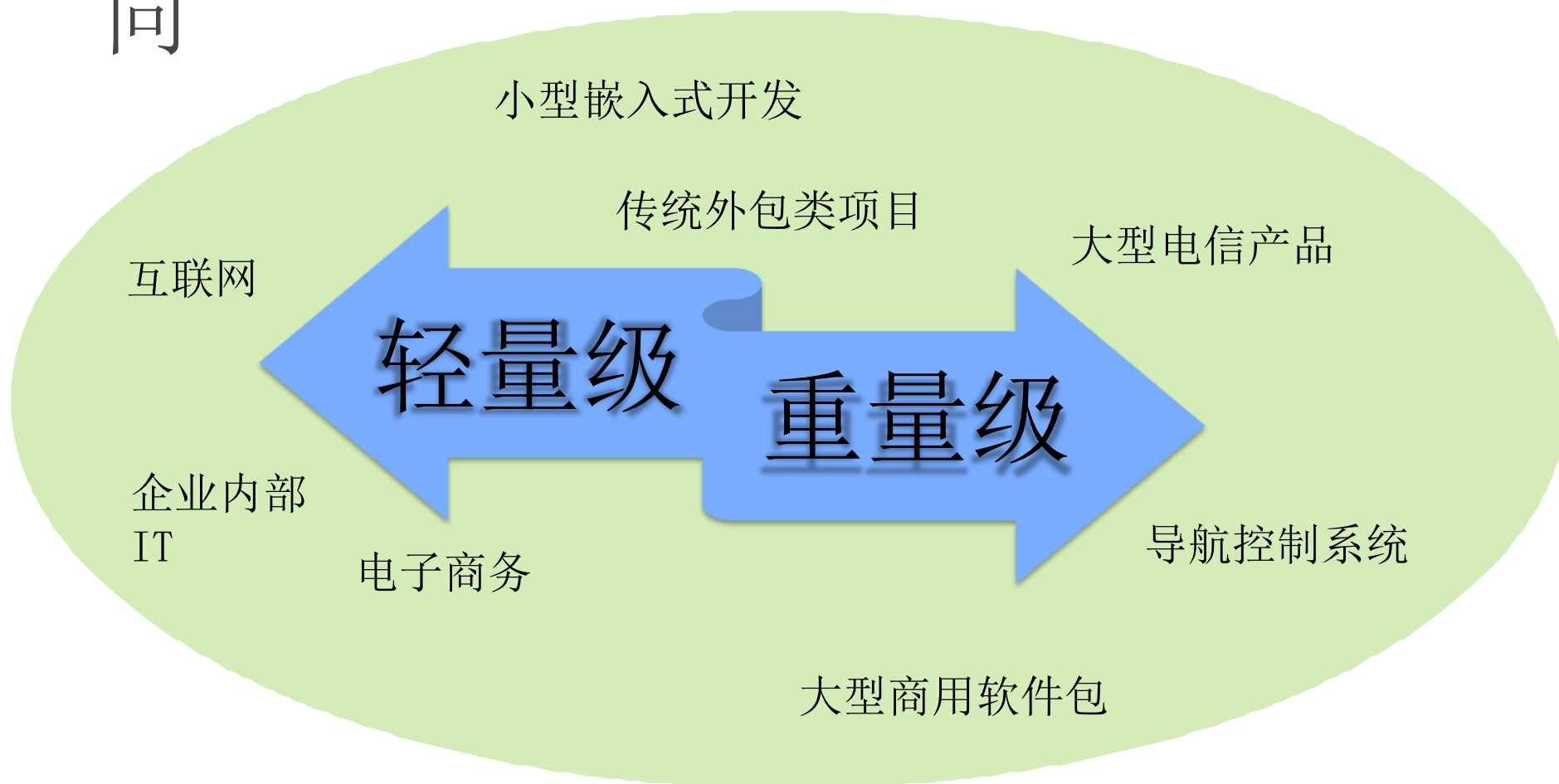
关注价值的创造，而非任务的完成



实践：个人 + 团队 + 组织

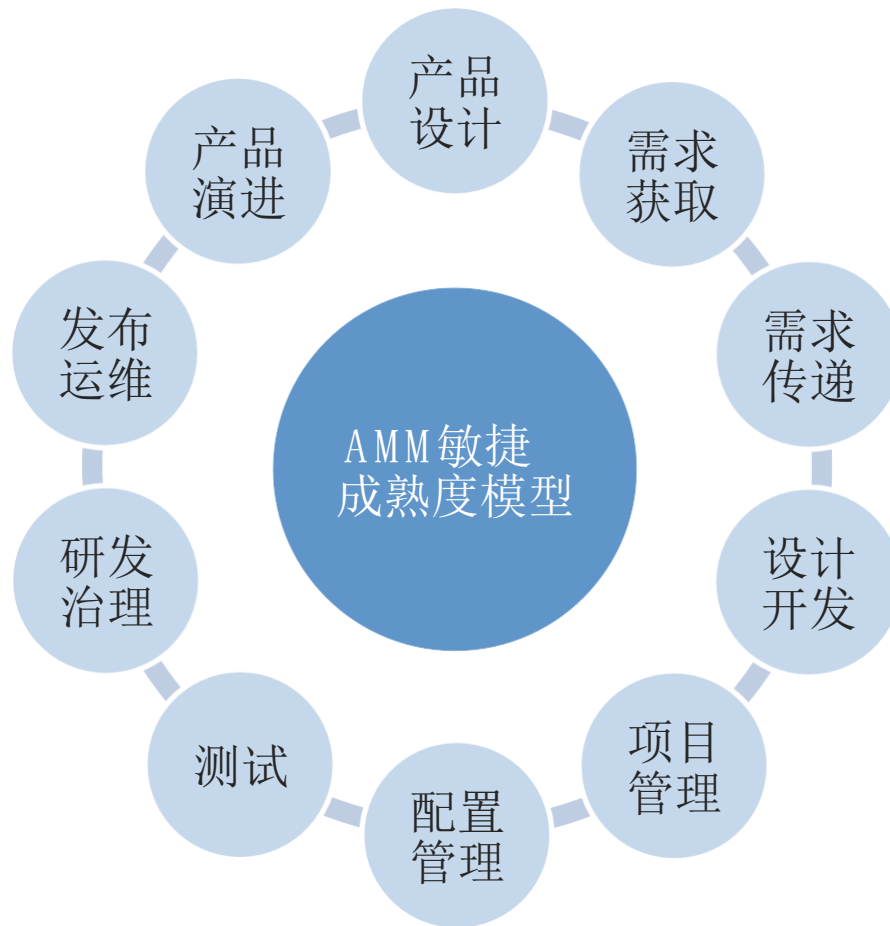


应用 - 产品类型不同



团队在透彻理解敏捷理念的基础上，可以灵活选择最适合自己的实践，避免教条化

应用 - 团队成熟度



敏捷宣言的演进

团队愿景和纪律

胜过

个体与交互

经过验证的学习

胜过

可用的软件

客户发现

胜过

客户协作

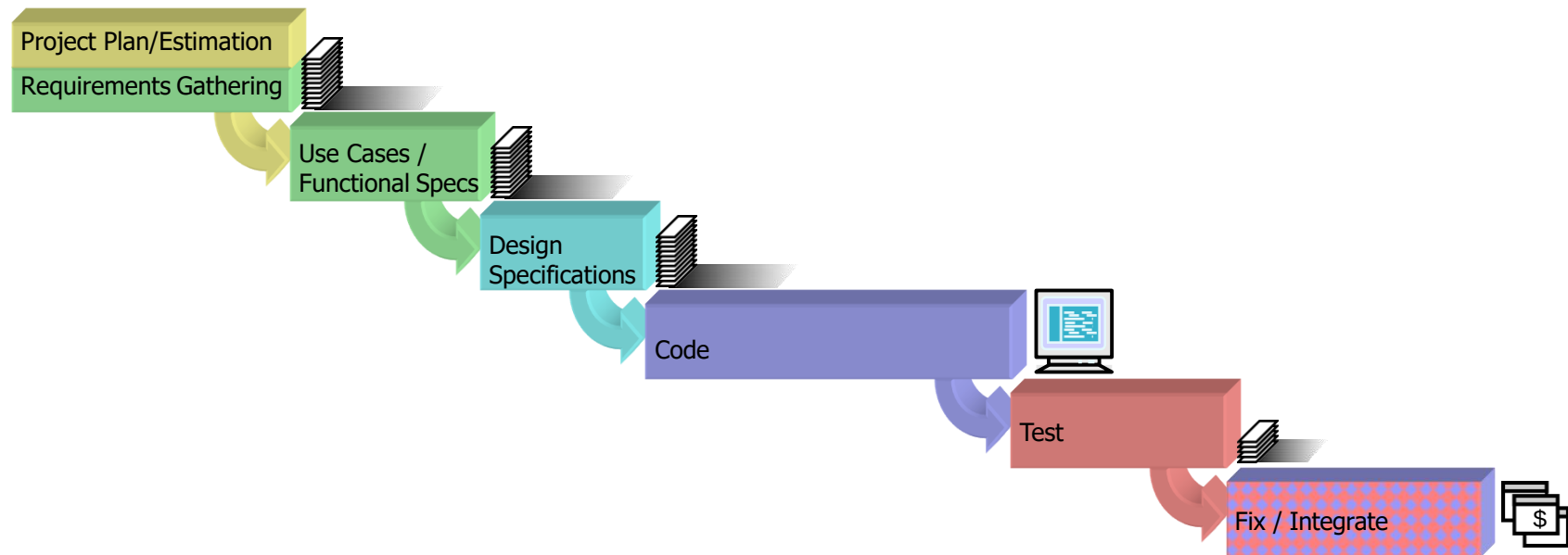
发起变化

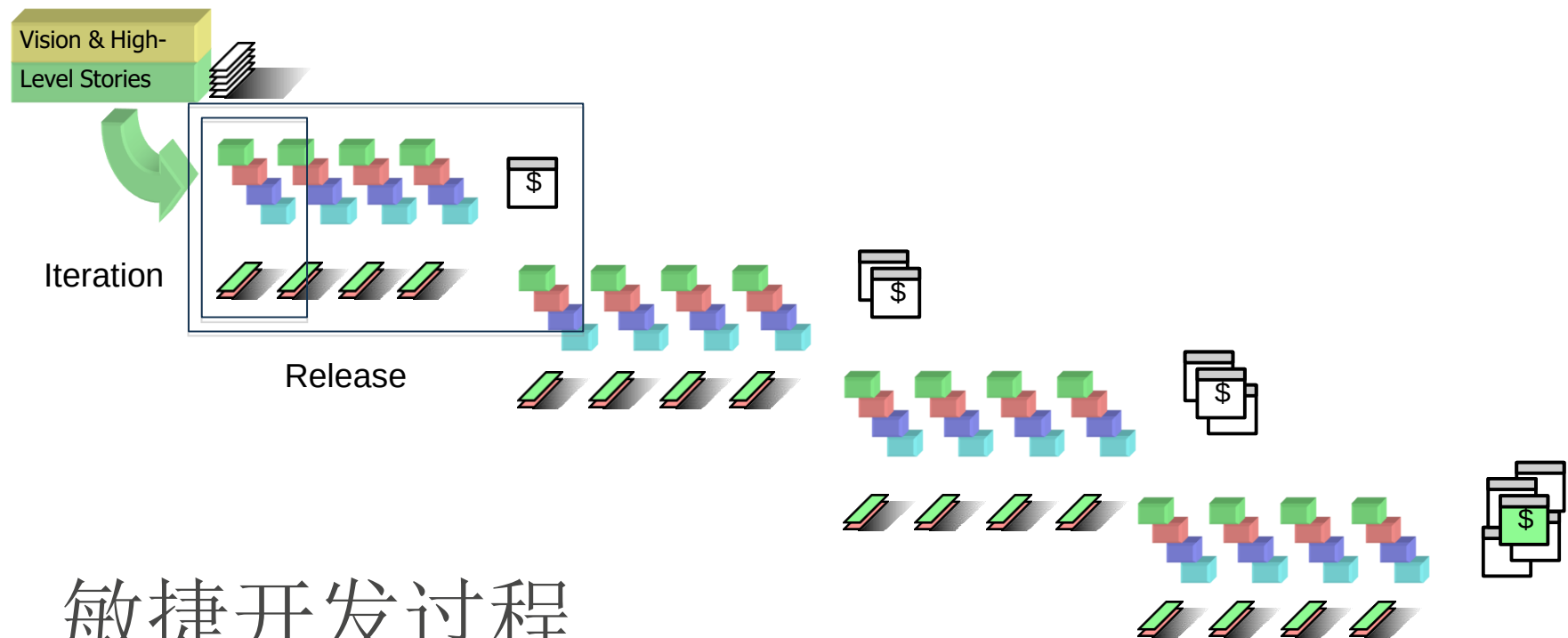
胜过

响应变化

迭代开发过程

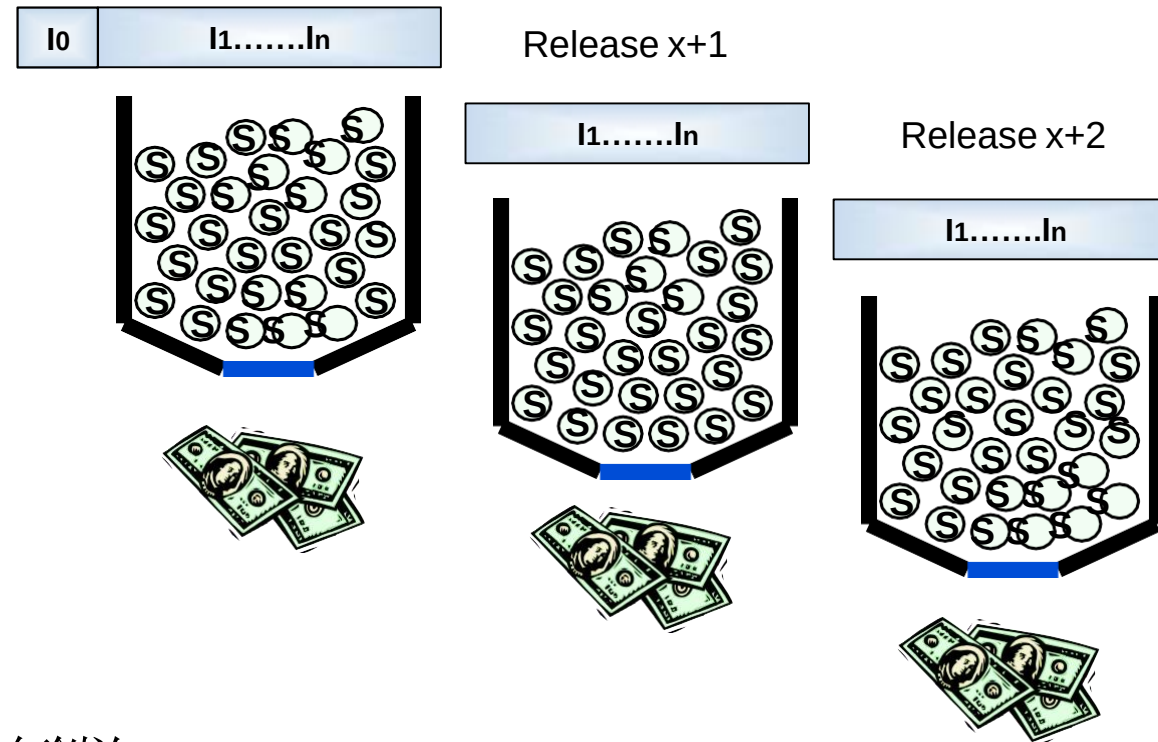
瀑布式开发





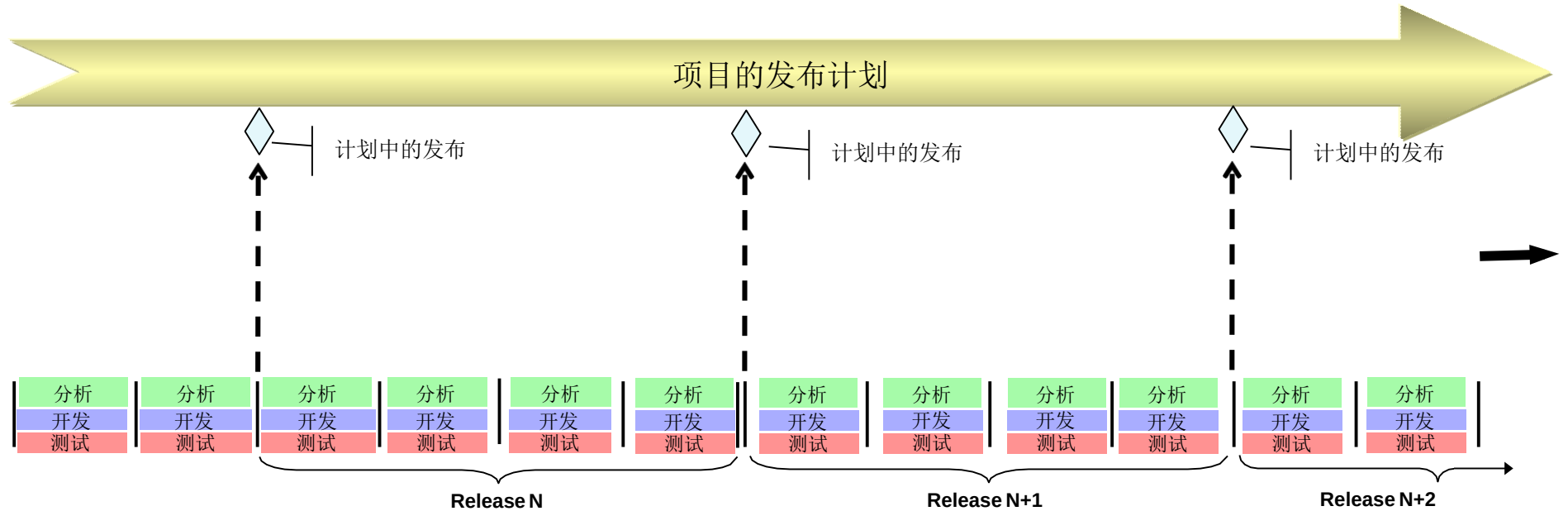
敏捷开发过程

发布计划 – 瓜分Stories



- ❖ 按优先级将故事装入每个发布
- ❖ 每个发布提供完整的业务价值
- ❖ 每个发布的历时可以不等

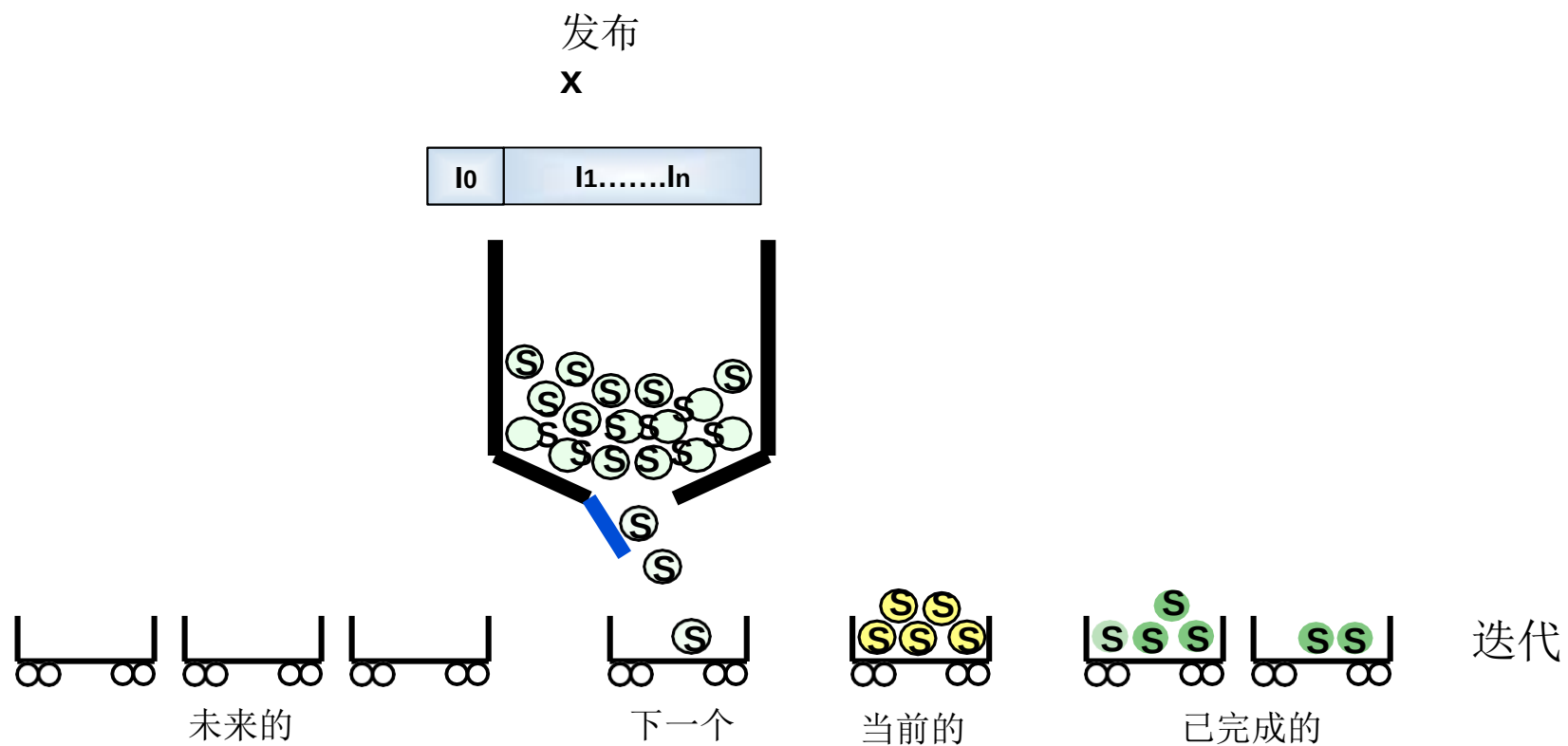
发布计划

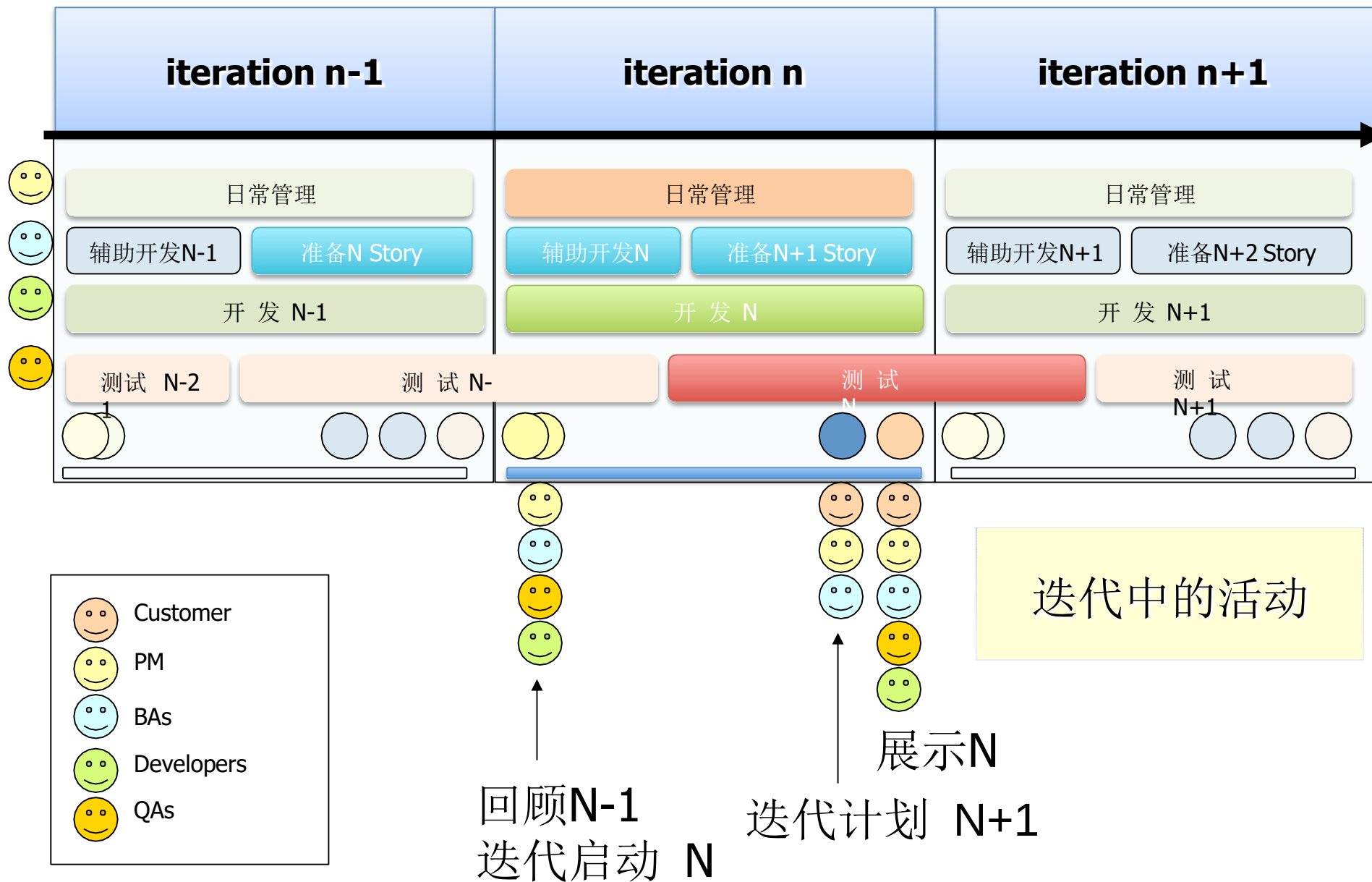


前提:

- ❖❖每个故事的大小 – 大小估计 Estimation
- ❖❖迭代长度
- ❖❖预计开发效率—每个迭代开发速度 Velocity

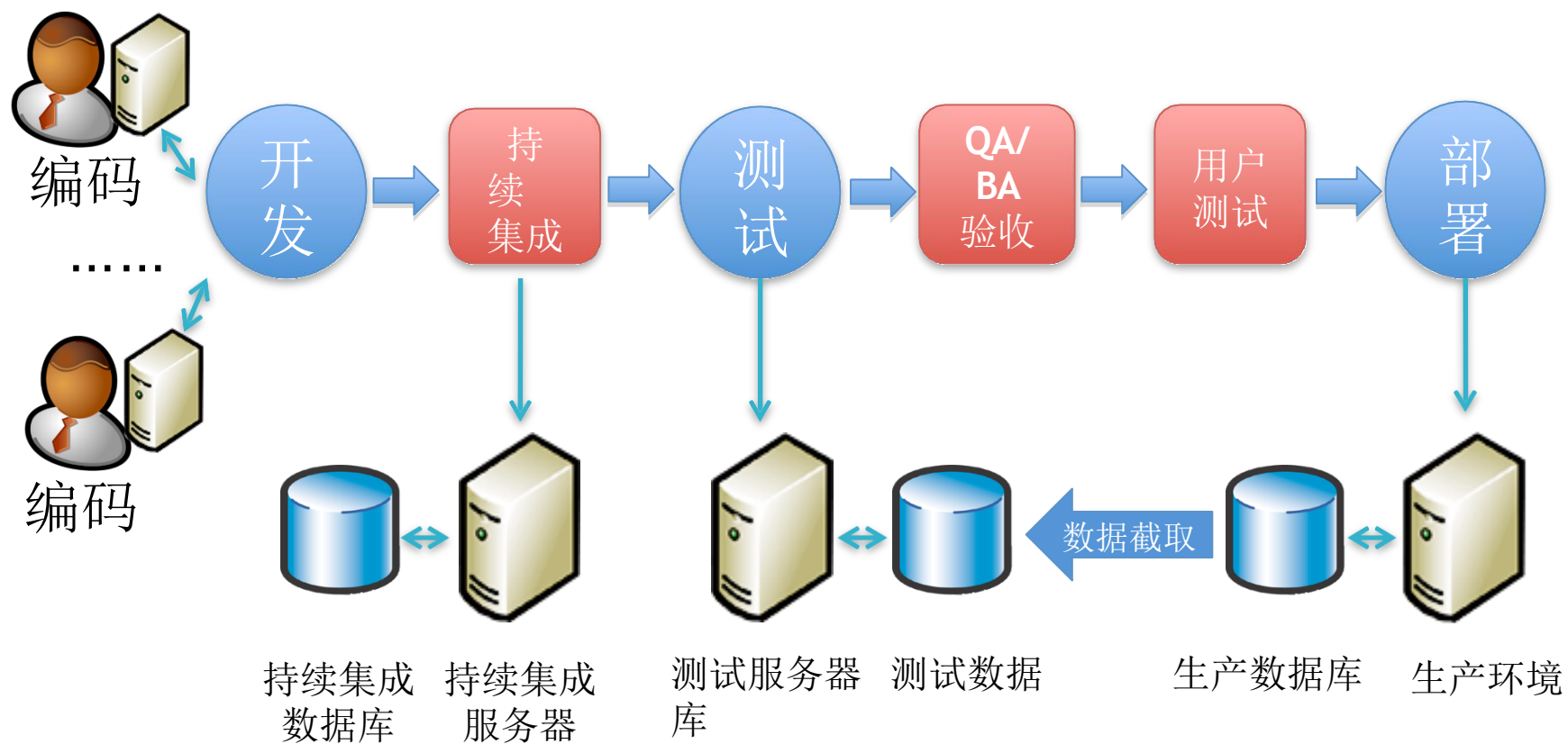
迭代计划





开发环境





Q&A

