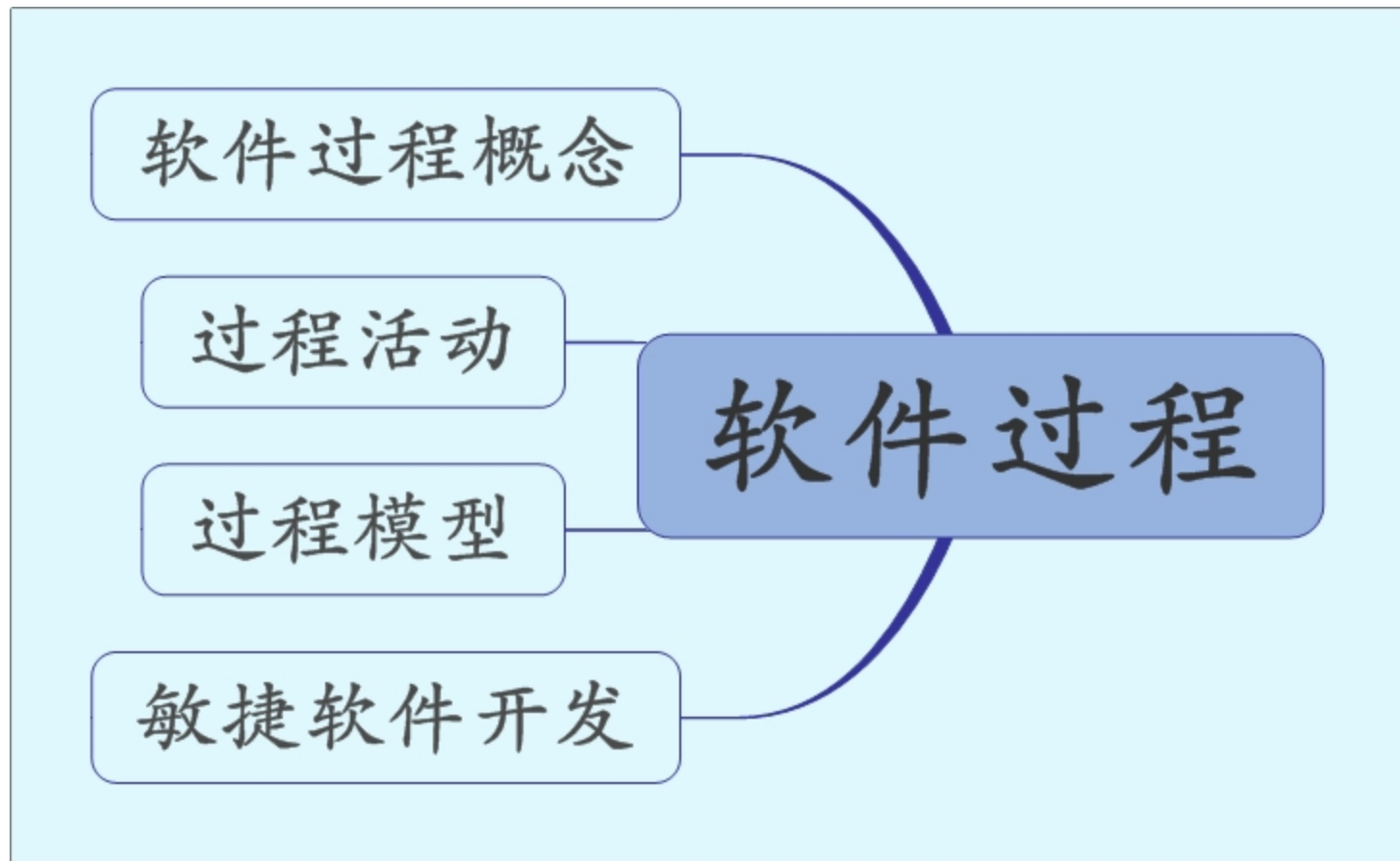


# outline



# 敏捷诞生的历史背景

20世纪60年代 软件作坊



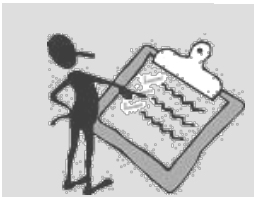
软件规模小，以作坊式开发为主

70年代 软件危机



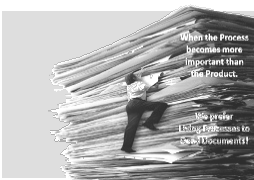
硬件飞速发展，软件规模和复杂度激增，引发软件危机

80年代 软件过程控制



以“过程为中心”分阶段来控制软件开发（瀑布模型），一定程度上缓解了软件危机

90年代 重型过程



软件开发过程日益“重型化”，开发效率降低、响应速度变慢

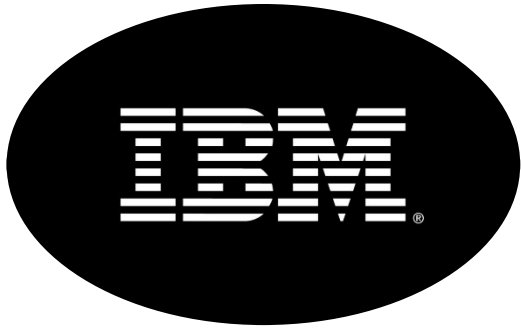
2001~今 敏捷正在流行



需求变化更快，交付周期成为企业核心竞争力，轻量级的，更能适应变化的敏捷软件开发方法被普遍认可并迅速流行

**重型过程** → **轻量型敏捷**

# 业界敏捷浪潮



# 敏捷宣言 (Agile manifesto)



- *We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*

- *Individuals and interactions* over *processes and tools*

个体和交互

过程和工具

- *Working software* over *comprehensive documentation*

可运行的软件

详尽的文档

- *Customer collaboration* over *contract negotiation*

客户合作

合同谈判

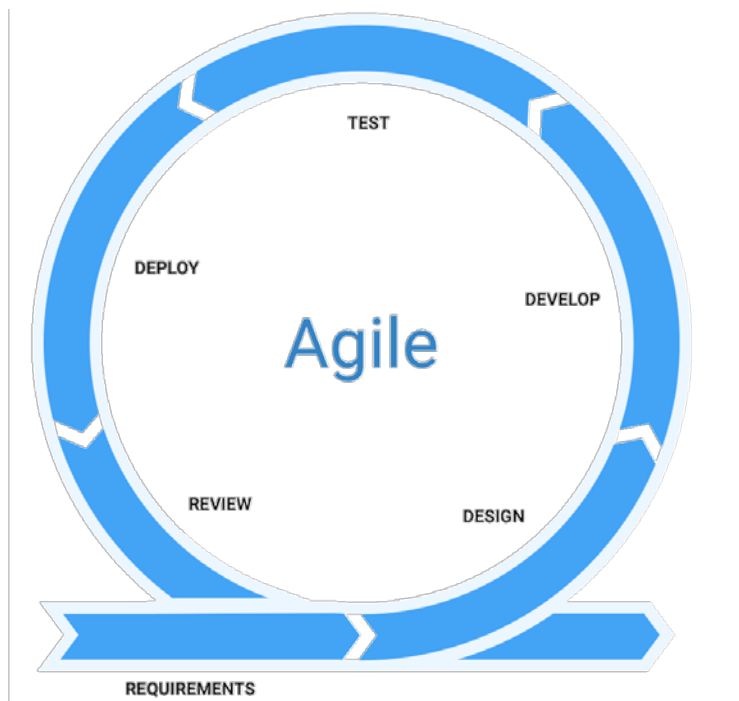
- *Responding to change* over *following a plan*

响应变化

遵循计划

- *That is, while there is value in the items on the right, we value the items on the left more.*

# 敏捷方法（Agile methods）



- **以人为核心、迭代、增量式**的开发方法
- 目标：“尽可能早地、持续地交付有价值的软件”使客户满意
- 灵活性
  - 用户能够在迭代周期的后期增加或改变需求
  - 尽快变更软件以满足变化的需求

# 敏捷方法的原则

## 客户参与

- 参与整个开发过程
- 提供新的系统需求
- 评估系统的每个增量

## 增量交付

- 用户确定每个增量中包含的需求
- 增量规模小
- 快速交付  
2~3 weeks

## 人胜于过程

- 开发团队成员的技术
- 不拘泥于特定的过程

## 拥抱变更

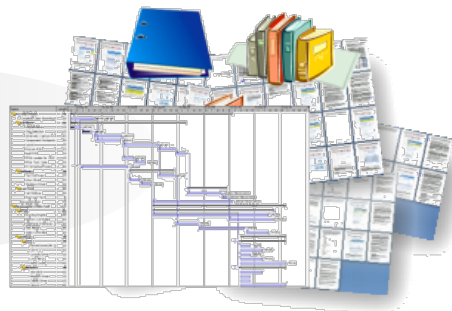
- 快速响应变更

## 保持简单

- 开发过程简单

# 敏捷方法的特征

传统开发

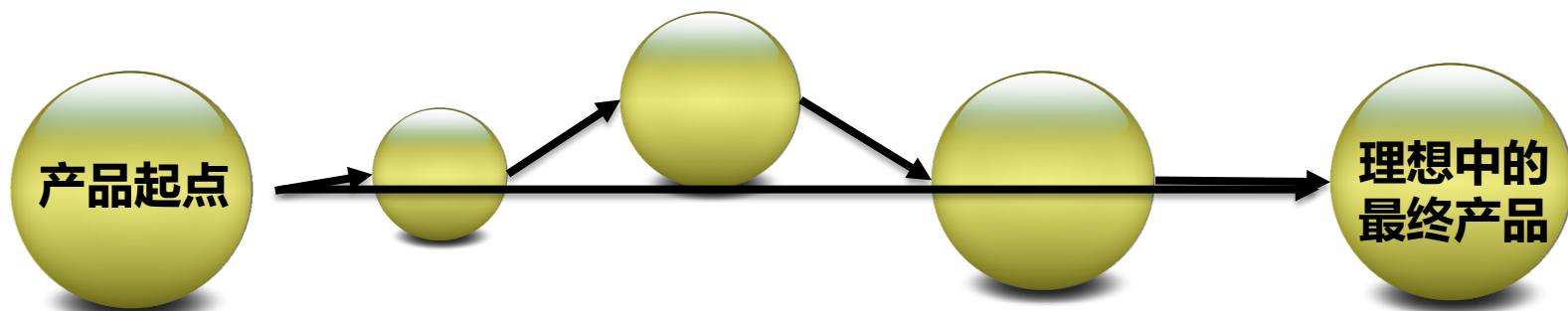
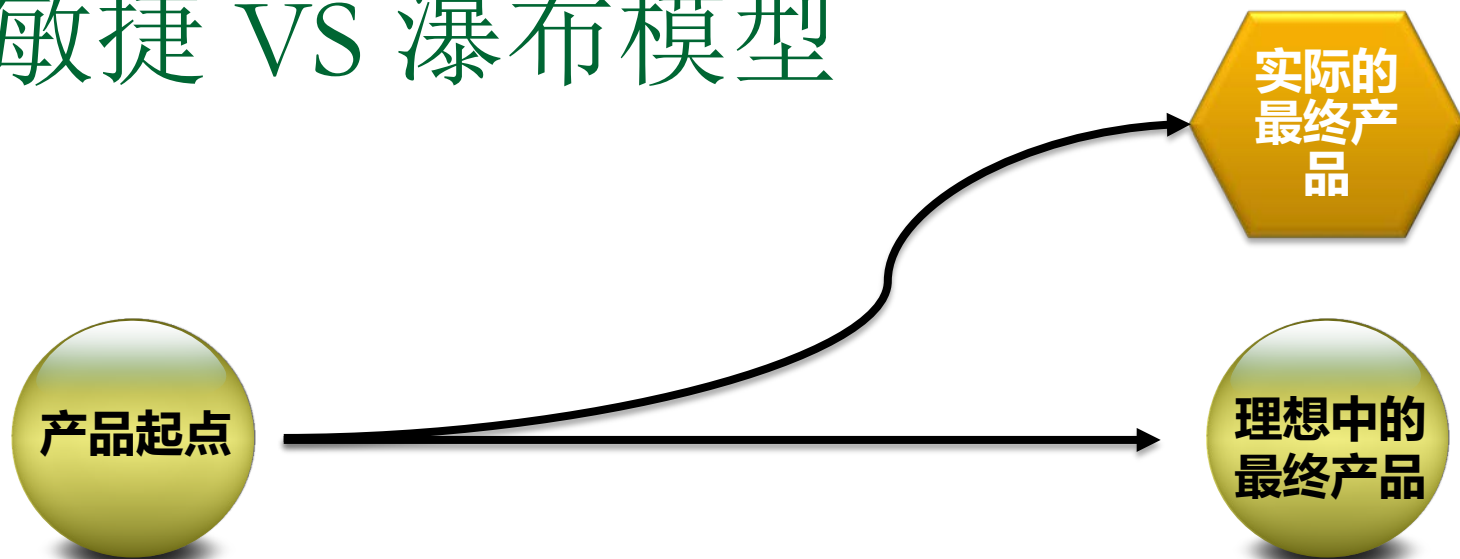


敏捷开发



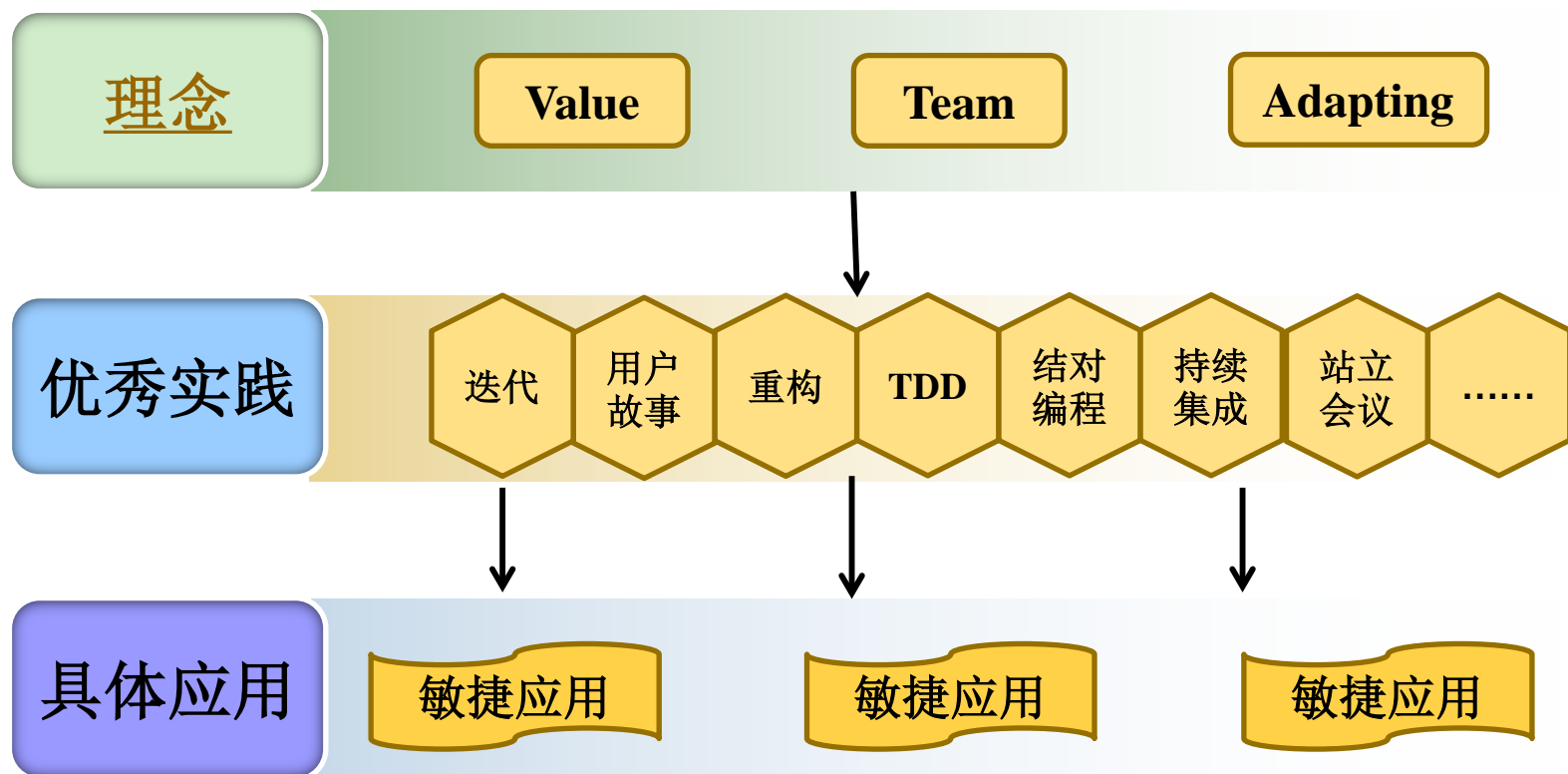
- 软件更像一个活着的植物，软件开发是自底向上逐步有序的生长过程，类似于植物自然生长
- 敏捷开发遵循软件客观规律，不断的进行迭代增量开发，最终交付符合客户需求的产品

# 敏捷 VS 瀑布模型





# 敏捷=理念+优秀实践+具体应用



敏捷包括3个层次 {

- 理念（敏捷核心思想）
- 优秀实践（敏捷的经验积累）
- 具体应用（能够结合自身灵活应用才是真正敏捷）

# 理念

## 聚焦客户价值 ( Value )

- “可工作的软件”
- “客户合作”

## 激发团队潜能 ( Team )

- 加强团队协作
- 提升沟通效率，  
面对面交流
- “个体和交互”

## 不断调整以适应变化 ( Adapting )

- 客户逐步发现真正的需求
- 需求变更
- “响应变化”



# 敏捷优秀实践

## 技术实践（极限编程XP）

- 用户故事
- 重构
- TDD
- 结对编程
- 持续集成

## 迭代开发

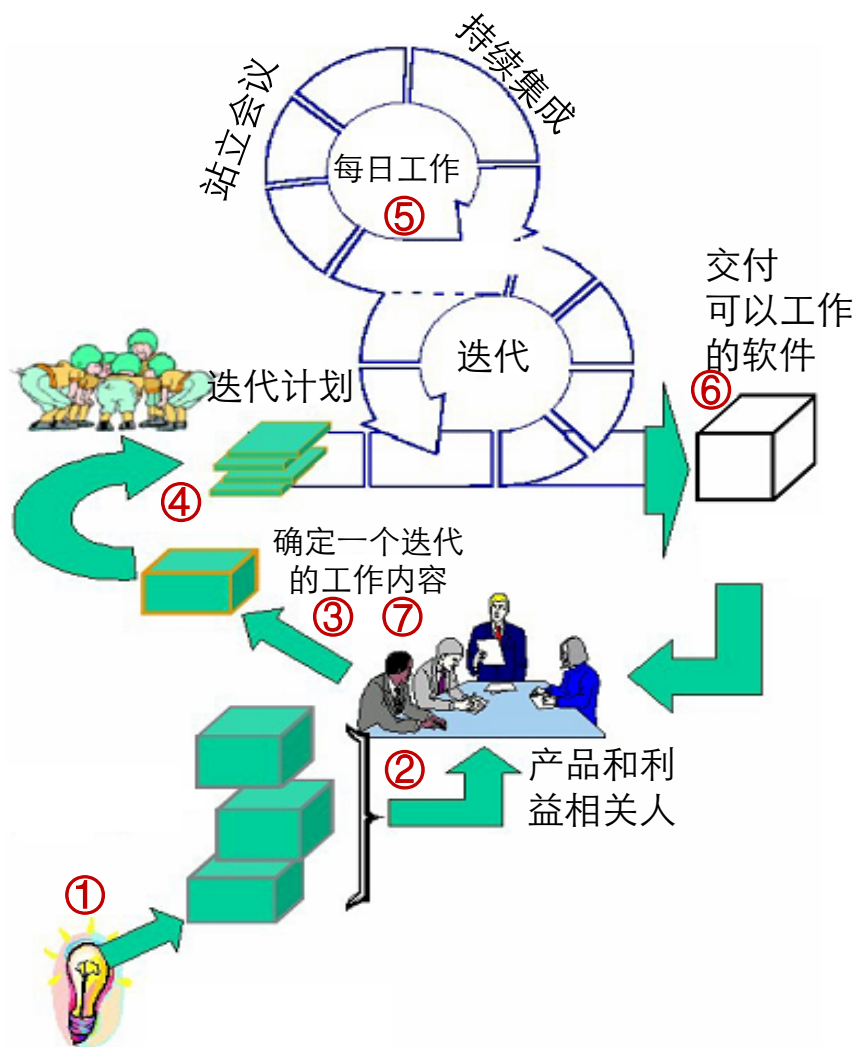
## 管理实践（Scrum）

- 迭代计划会议
- 每日站立会议
- 可视化管理
- 迭代验收

## 团队（Scrum）

- 敏捷团队角色
  - Product Owner（产品负责人）
  - Scrum Master（Scrum教练）
  - Team
- 完整团队实践

# 敏捷实施典型场景



- 1、PO和开发团队对产品业务目标形成共识
- 2、PO建立和维护产品需求清单（Backlog），并进行优先级排序
- 3、PO每轮迭代前，检验需求清单，并筛选高优先级需求进入本轮迭代开发
- 4、开发团队细化本轮迭代需求，并按照需求的优先级，依次在本轮迭代完成
- 5、开发团队每日站立会议、持续集成，使开发进度真正透明
- 6、PO对每轮迭代（2 - 4周）交付的可工作软件进行现场验收和反馈
- 7、回到第3步，开始下一轮迭代