

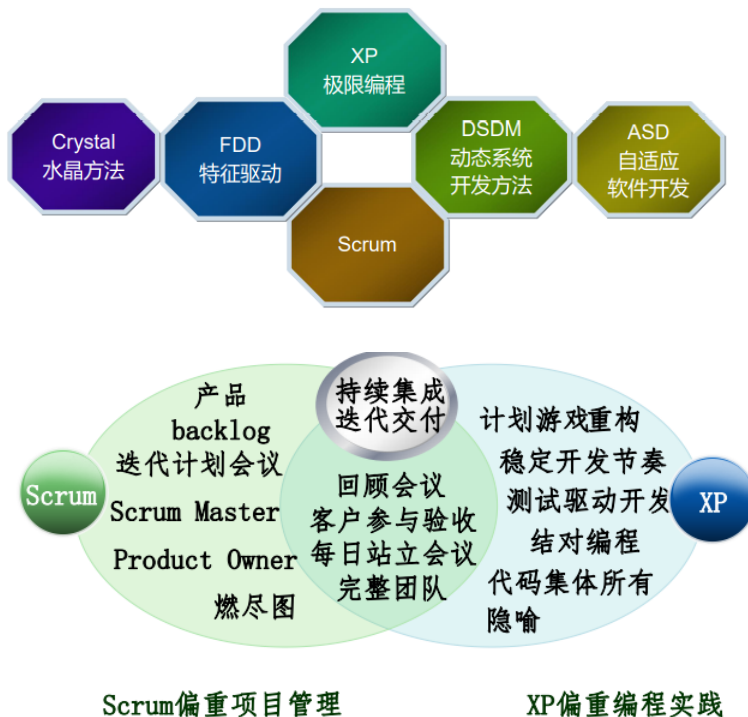
## 6.1 敏捷开发

- 敏捷方法概述

- 软件开发之道

- 如今软件开发更应该侧重于
      - 弹性的开发管理方式
      - 通过初始计划开始工作
      - 项目的资源管理和控制
    - 交付的价值
      - 高质量的交付物
      - 系统不是一次构建而成，而是迭代演进的
      - 基于完整的场景构建计划，并按优先级执行
    - 互联网时代的软件开发
      - 快鱼吃慢鱼
      - 版本发布成本低
      - 追求创新
      - 需要快速响应用户变化
      - 需求不确定性高
      - 关注用户行为

- 敏捷方法与敏捷宣言



敏捷开发方法是一组轻量级开发方法的总称，包含很多具体的开发过程和方法，最有影响的两个方法是极限编程（XP）和Scrum开发方法。

- 敏捷开发

- 敏捷开发是一种基于更紧密的团队协作、能够有效应对快速变化需求、快速交付高质量软件的迭代和增量的新型软件开发方法
- 更关注协作、质量、可工作的产品、全才化的专才、基于实践而非理论
- 适应而非预测
- 以人为导向而非以过程为导向

- 敏捷开发宣言

- 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
- 最高目标——尽早且持续的**交付有价值的软件**
- **欢迎对需求提出变更**
- 不断**交付**可用软件，周期从几周到几个月，**越短越好**
- **善于激励**项目人员
- 业务人员与开发人员**一起工作**
- **面对面的交谈**
- **可用的软件**是**衡量进度**的主要指标
- **可持续的开发速度**
- 追求**技术卓越和良好设计**
- 做到**简单**
- **自组织的团队**
- **定期反省**如何能做到最好，**相应的调整**团队行为

- 敏捷开发核心理念

- 聚焦客户价值
  - 消除开发中的浪费
  - 随时构建质量，不容忍缺陷
  - 交付刚刚好的系统
  - 及时消除技术债务，持续保持快速响应
- 激发团队潜能
- 不断调整以适应变化

- 敏捷开发方法的应用

- Scrum 框架



## • Scrum 框架介绍

- 是一种兼顾计划性与灵活性的敏捷开发过程，它将整个开发过程分为若干次更小的迭代，每个迭代周期称为一个冲刺（Sprint）
- 将整个软件生命周期分成若干个小迭代，每一次迭代就是一个小的瀑布模型，包括需求分析、设计、实现和测试，结束时需要生成一个稳定和被验证过的软件版本
- 关键点
  - 每一次迭代建立在稳定的质量基础上，并作为下一轮迭代的基线，系统功能随迭代稳定增长和不断完善
  - 每次迭代邀请用户代表验收，提供需求是否满足的反馈
  - 在一次迭代中，一旦团队做出承诺，就不允许变更交付件和交付日期，入股欧发生重大变化，产品负责人可以终止当次迭代
  - 迭代中可能会出现“分解”和“澄清”，但不允许添加新工作或对现有工作进行“实质变更”
  - 对于“分解”和“澄清”，如果存在争议，那么将其认定为变更，放到产品订单中下一次迭代再考虑

## • Scrum 团队角色

角色	职责	说明
产品负责人	<ul style="list-style-type: none"> <li>代表利益相关人（如用户、市场销售、高层管理者等）对产品的投资回报负责</li> <li>确定产品发布计划</li> <li>定义产品需求并确定优先级</li> <li>验收迭代结果，并根据验收结果和需求变化更新需求清单和优先级</li> </ul>	除了负责客户需求外，内部任务如重构、持续集成环境搭建等也由产品负责人纳入统一管理
Scrum主管	<ul style="list-style-type: none"> <li>辅导团队正确应用敏捷实践</li> <li>引导团队建立并遵守规则</li> <li>保护团队不受干扰</li> <li>推动解决团队遇到的障碍</li> <li>激励团队</li> </ul>	不命令和控制开发团队
开发团队	<ul style="list-style-type: none"> <li>负责估计工作量并根据自身能力找出最佳方案去完成任务且保证交付质量</li> <li>向产品负责人和利益相关人演示工作成果（可运行的软件）</li> <li>团队自我管理、持续改进</li> </ul>	<ul style="list-style-type: none"> <li>由5-9名跨功能领域人员组成</li> <li>集中在一起工作</li> <li>共同目标，共担责任</li> <li>严格遵守团队规则</li> </ul>

- 产品负责人
  - 职责：定义开发密保以及需要实现的特性和优先级
- Scrum 主管
  - 职责：保证团队高效而不受干扰地工作，优化工作条件和过程
- 团队成员
  - 职责：自组织地完成项目开发，使用一切可行手段保证进度和质量
- 团队组织

- 民主式结构：小组成员完全平等，名义上的组长与其他成员没有任何区别；大家享有充分的民主，项目工作由全体讨论协商决定，并根据每个人的能力和经验进行适当分配。

- 优点：同等项目参与权激发大家创造力，有利于攻克技术难关
- 缺点：缺乏明确的权威领导，很难解决意见分歧

- 全功能的整体团队



- Scrum 制品和活动

- 制品

- 产品订单

- 从客户价值角度理解的产品功能列表
- 功能、缺陷、增强都可以是产品订单项
- 整体上从客户价值进行优先级排序

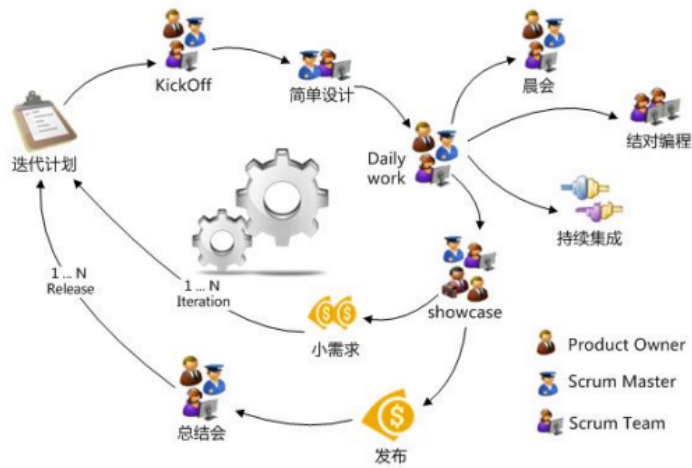
- 迭代订单

- 从开发技术角度理解的迭代开发任务
- 简单环境：产品订单项直接分配到迭代中
- 复杂环境：把产品订单项分成若干开发任务
- 迭代过程中迭代订单不可修改

- 可工作软件

- “可交付”应视不同情况前提设定和选定交付标准
- 正式产品可能包括使用文档，在新产品开发初期可能只需要交付勉强看到效果的产品

- 活动



- 迭代规划会议

- 在每次迭代开始时召开，一般不超过8小时，目的是选择和估算本次迭代的工作项
- 分为两部分
  - 第一部分以需求分析为主，选择和排序本次迭代需要实现的订单条目
  - 第二部分以设计为主，确定系统设计方案和工作内容
- 第一部分
  - 会议目的
    - 了解用户需求
    - 产品负责人准备：条目化的需求、优先级排序、最近迭代希望看到的功能
  - 基本要求
    - 估算和排序的产品订单列表
    - 计划表、重要人员详细联系信息
    - 参会成员：团队成员、主管、产品负责人
  - 会议过程
    - 从第一个产品订单Product Backlog条目（即用户故事）开始，讨论该条目，以深入理解。
    - 分析并明确用户验收测试
    - 找出非功能性需求（性能、稳定性、.....）
    - 找出验收条件
    - 弄清楚需要“完成”到何种水平
    - 获得订单条目各个方面的清晰了解
    - 绘制所需交付物的相关图表，包括流程图，UML图，手绘草稿、屏幕界面设计等
    - 回到步骤1，选取下一个产品订单条目
  - 会以输出

- 选择好的 Product Backlog 条目
  - 各个 Backlog 条目的需求
  - 各个条目的用户验收测试
  - 注意不要改变订单条目大小，不要估算任务
- 流程检查
  - 团队讨论认为能完成多少工作
  - 询问能够完成的订单条目
- 第二部分
  - 会议过程
    - 从第一个Backlog条目开始，查看挂图，确定对于客户需求理解正确
    - 从第一个Backlog条目开始，查看挂图，确定对于客户需求理解正确
      - 我们需要编写什么样的接口？
      - 我们需要创建什么样的架构？
      - 我们需要更新哪些表？
      - 我们需要更新或是编写哪些组件？
    - 当团队明确知道自己应该如何开发该功能后，就可以转向下一个条目
    - 在会议的最后10分钟，团队成员使用即时贴写出初步的任务。这样可以帮助团队成员知道接下来的工作从哪里开展，将这些任务放在任务板上
  - 会议输出
    - 应用设计、架构设计图、相关图表
    - 确保团队知道如何完成任务
- 每日站立会议
  - 目的
    - 做计划、协调每日活动，报告讨论障碍
    - 更新任务板和燃尽图
  - 基本要求
    - 任务版、即时贴、马克笔
    - 成员：团队、Scrum主管
    - 每天15分钟，相同时间和地点
    - 团队成员在聆听他人发言时，应该想：我该怎么帮他做得更快？
    - 准时出席，不要超出限制时间
    - 不要讨论技术问题，不要转变会议话题
    - Scrum主管不要替团队成员移动任务卡片，不要替团队更新燃尽图
    - Scrum主管不要提出问题，团队成员不要向Scrum主管或者管理层人员报告
  - 会议过程

- ①团队聚在故事板旁边，可以围成环形。
- ②从左边第一个开始，向团队伙伴说明他到现在完成的工作。
- ③该成员将任务板上的任务放到正确的列中。
- ④可以的话，该成员可以选取新的任务，将其放入“进行中工作”列。
- ⑤如果该成员遇到问题或障碍，就要将其报告给 Scrum主管。
- ⑥每个团队成员重复步骤②到步骤⑤。



- 回答问题

上次例会后完成了什么？

遇到了什么困难（或障碍）？

下次例会前计划做什么？

- 会议输出

团队彼此明确知道各自的工作，最新的工作进展图

最新的“障碍Backlog”

最新的“Sprint Backlog”

- 迭代总结会议

- 每一次迭代完成后，都会举行一次迭代总结会议，会上所有团队成员都要反思这个迭代。举行迭代总结会议是为了进行持续过程改进，会议的时间限制在4小时

- 要点

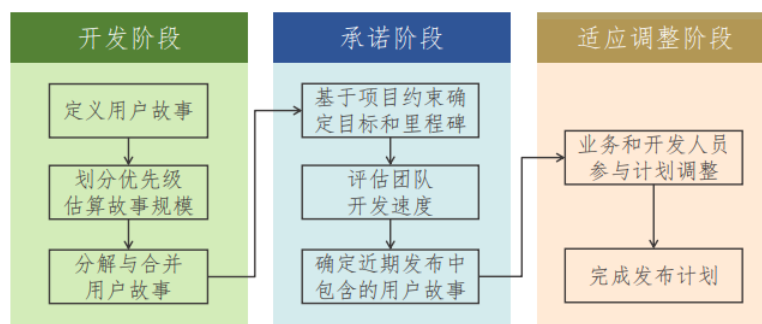
- 会议气氛：团队全员参加，气氛宽松自由，畅所欲言，头脑风暴发现问题，共同分析根因；
- 关注重点：共同讨论优先级，将精力放在最需要的地方（关注几个改进足以）
- 会议结论要跟踪闭环：可以放入迭代订单中

- 敏捷规划与可视化管理

- 规划

- **实行两级项目规划。**将面向整个项目范围的宏观计划和面向当前下一次迭代的微观计划相结合,渐进明细地进行项目规划。

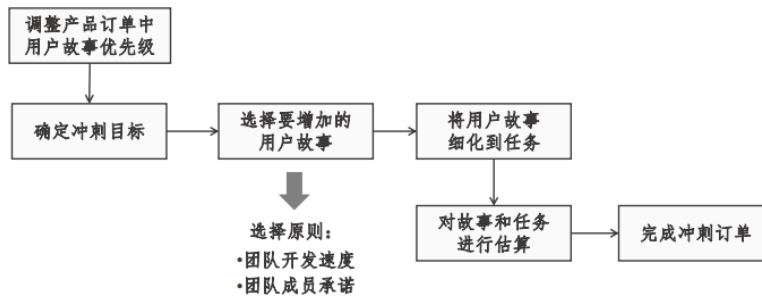
- 发布规划



- 是对整个产品发布过程的展望，其结果是产生产品订单
- 定义用户故事并进行优先级划分
- 指定发布计划
- 估算规模以及团队开发速度



- 迭代规划



- 只是对一次迭代的展望，其结果是确定包含一次迭代中具体工作任务的迭代订单
- 确定迭代目标并选择用户故事
- 用户故事分解和细化到任务
- 对故事和任务进行实践估算

- 可视化管理

- 任务白板是团队开发的晴雨表，它将团队的任务和进度可视化地展现出来。而引入电子白板可能会削减团队之间的沟通，降低团队的透明度，违背了敏捷重视人和团队的原则。
- 燃尽图：以图形化方式展现了剩余工作量（Y轴）与时间（X轴）的关系

