

# CS10102302

# 敏捷开发方法

赵君峤 副教授

计算机科学与技术系 电子与信息工程学院

同济大学

# 教学提纲

1	敏捷方法概述
2	Scrum框架
3	敏捷开发实践

# 教学提纲

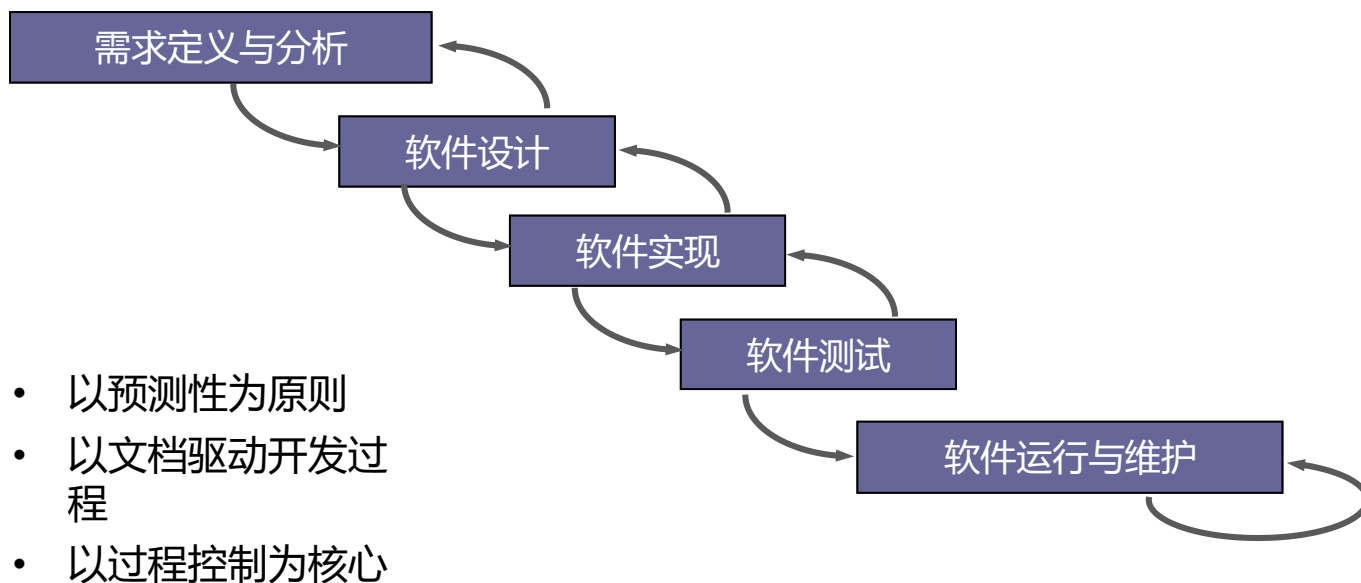
1

## 敏捷方法概述

- 软件开发之道
- 敏捷方法与敏捷宣言
- 敏捷开发核心理念
- 敏捷开发方法的应用

# 传统的软件开发模式

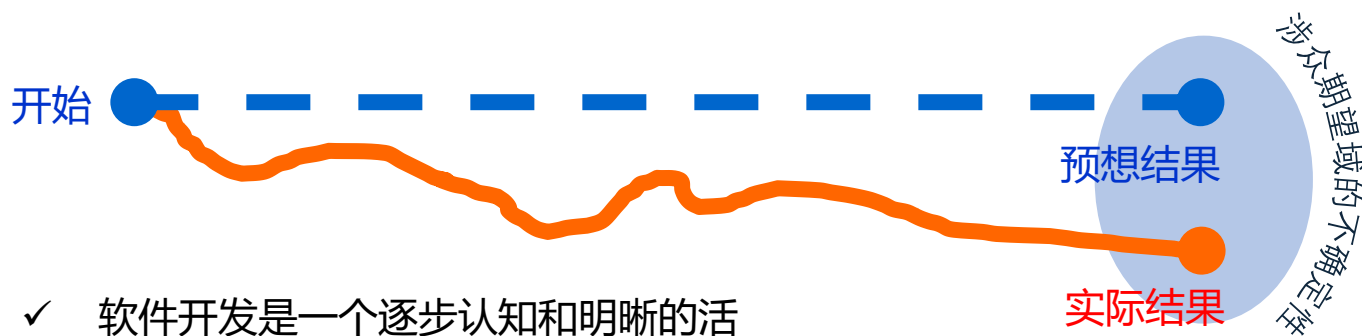
瀑布模型是最典型的预见性开发方法，严格遵循预先计划的需求分析、设计、编码、集成、测试、维护的步骤顺序进行。



# 软件开发之道

软件开发是否可以实现一个完整、详尽的计划？软件项目能否预先考虑到所有的风险？

软件项目中难以预知所有的内容和风险！！！！



- ✓ 软件开发是一个逐步认知和明晰的活动
- ✓ 软件开发中的变化是实际存在和必然的

# 软件开发之道

今天的软件开发过程更应侧重于

- 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



动极而静，静极复动  
一动一静，互为其根

# 软件开发之道

质量就是软件产品对于某个（或某些）人的价值。



正确的  
软件



一个软件要能够满足用户的需求，为用户创造价值。这里的价值可以体现在两个方面，即为用户创造利润和减少成本。

软件运行  
正确



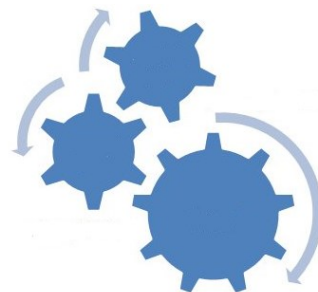
软件没有或者有很少缺陷，具有很强的扩展性、良好的性能以及较高的易用性等。

# 软件开发之道

您是想获取一些更有价值的交付产品呢，  
还是只想完成进度表!!

## 软件开发应更关注于交付的价值

- ✓ 高质量的交付物是最重要的
- ✓ 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

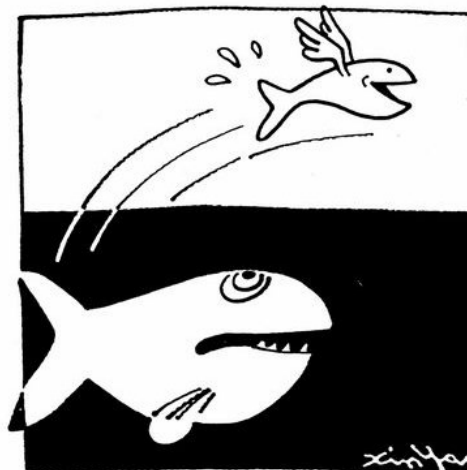




# 互联网时代的软件开发

## 互联网产品的开发特点

- 快鱼吃慢鱼
- 版本发布成本很低
- 追求创新
- 需要快速响应用户的变化
- 需求不确定性高
- 关注用户行为



如今不是大鱼吃小鱼,而是快鱼吃慢鱼。

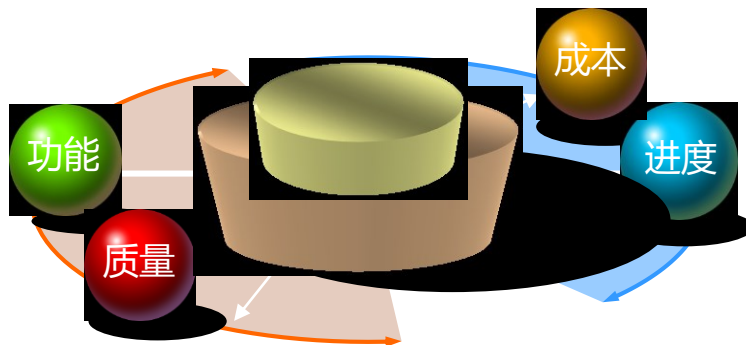
好的架构（产品）是**长出来的**，而不是设计出来的

# 敏捷开发方法



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。

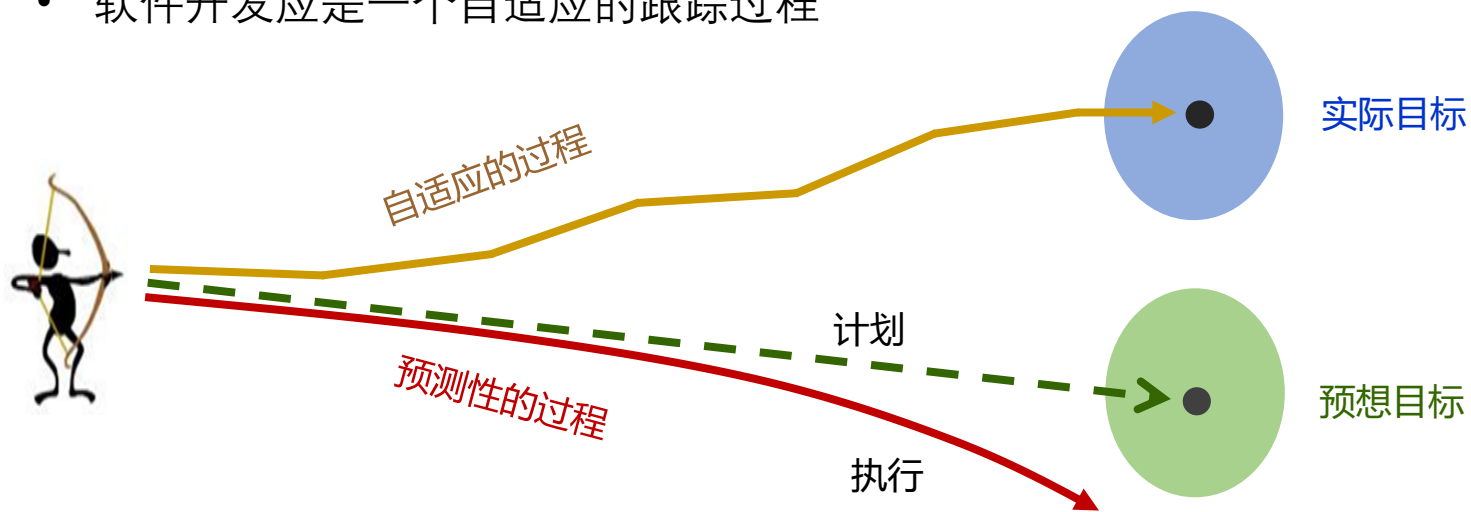
- 更关注协作
- 更关注质量
- 更关注可工作的产品
- 更关注全才化的专才
- 基于实践而非基于理论



# 敏捷开发方法

## 敏捷方法：适应而非预测

- 需求是不可预测的
- 软件开发应是一个自适应的跟踪过程



# 敏捷开发方法

敏捷方法：  
以人为导向而非过程导向

- 软件开发过程中，人的因素是第一位的
- 强调软件开发中相关人员间信息交流

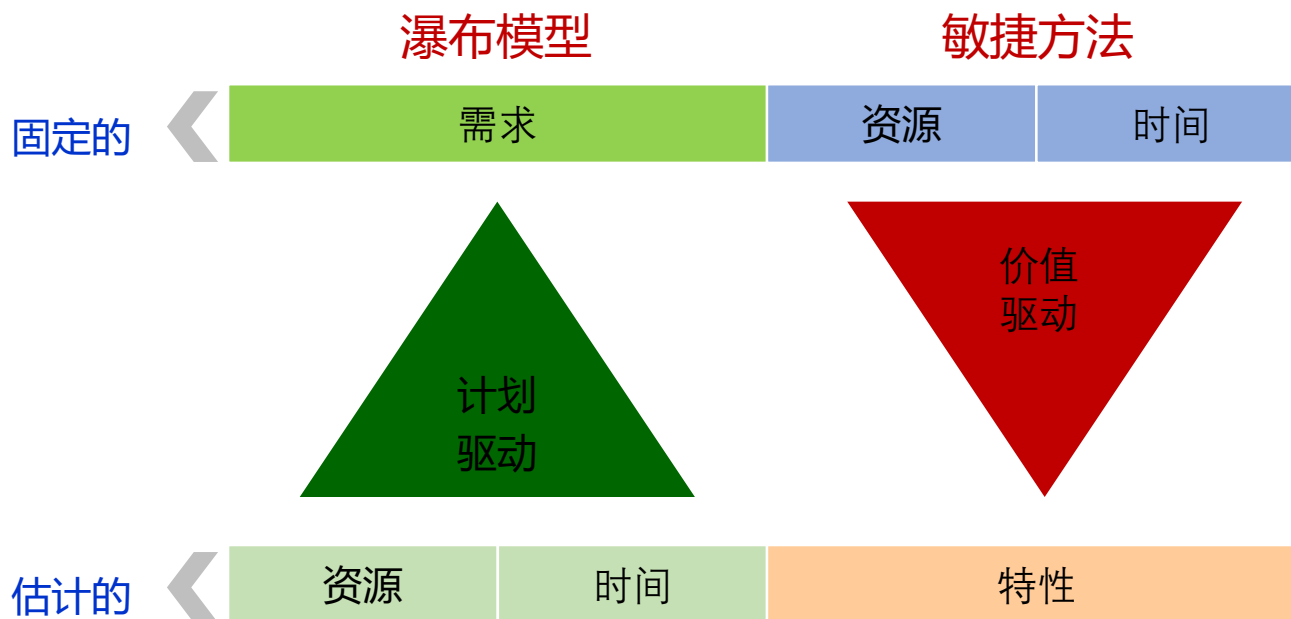
项目失败的原因最终都可追溯到某个信息没有及时准确地传递到应该接收它的人。

—— *Alistair Cockburn*

人特别擅长面对面的交流，面对面交流的成本要远远低于文档交流的成本。

—— *Alistair Cockburn*

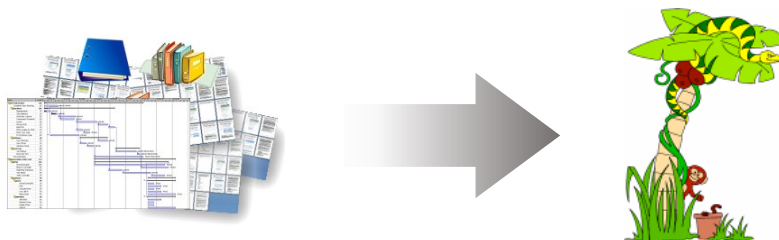
# 敏捷开发方法



# 敏捷开发方法

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。敏捷方法遵循软件的客观规律，不断进行迭代式增量开发，最终交付符合客户价值的产品。

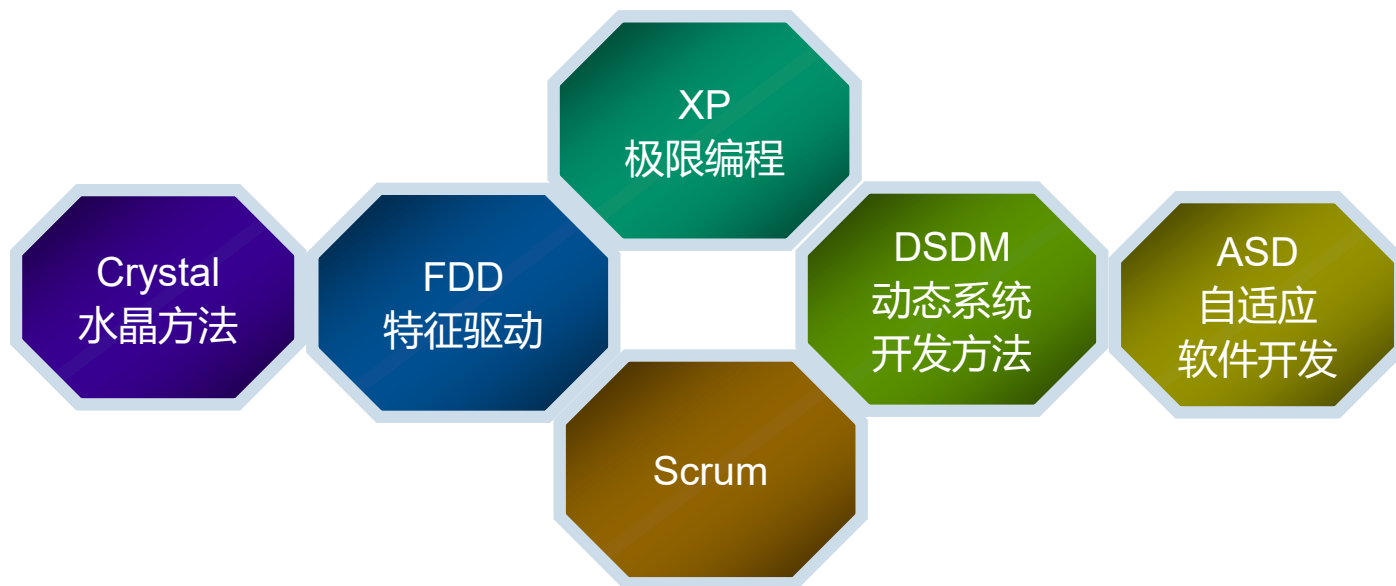
传统开发方法



敏捷开发方法

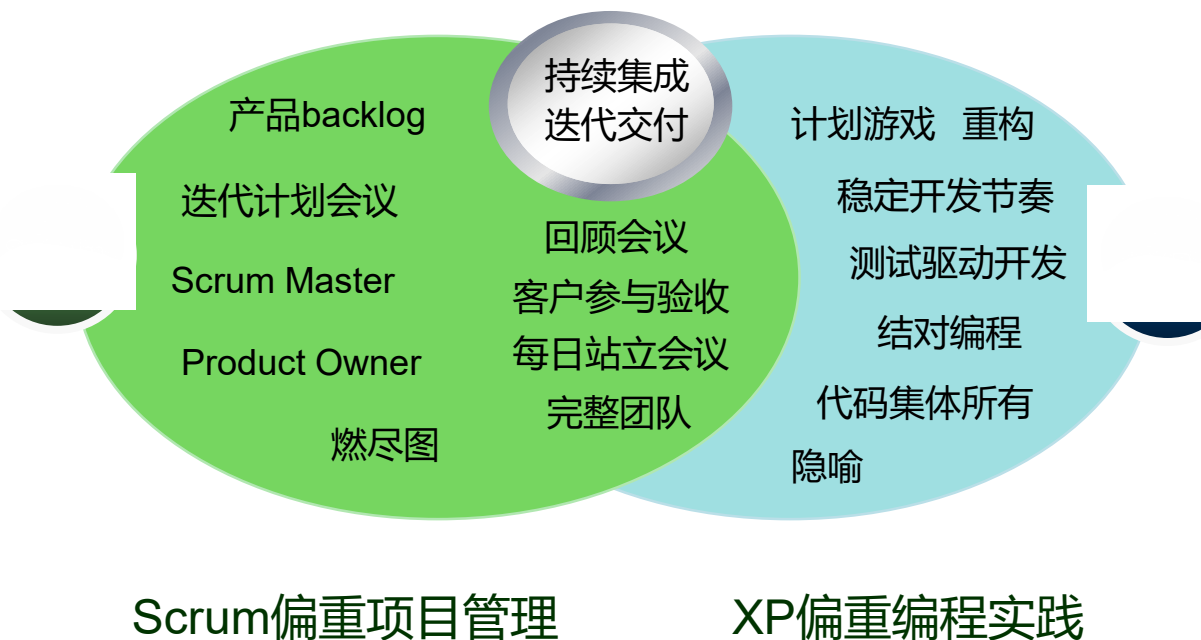


# 敏捷开发方法



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。

# 敏捷开发方法





# 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

可以工作的软件

客户合作

响应变化

胜过

胜过

胜过

胜过

过程和工具

面面俱到的文档

合同谈判

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。



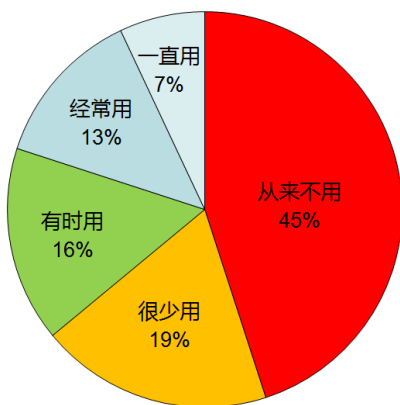
# 敏捷宣言

- 我们的最高目标是，通过尽早和持续地交付有价值的软件来满足客户。
- 欢迎对需求提出变更——即使是在项目开发后期。要善于利用需求变更，帮助客户获得竞争优势。
- 要不断交付可用的软件，周期从几周到几个月不等，且越短越好。
- 项目过程中，业务人员与开发人员必须在一起工作。
- 要善于激励项目人员，给他们以所需要的环境和支持，并相信他们能够完成任务。
- 无论是团队内还是团队间，最有效的沟通方法是面对面的交谈。
- 可用的软件是衡量进度的主要指标。
- 敏捷过程提倡可持续的开发速度，项目方、开发人员和用户应该能够保持恒久稳定的进展速度。
- 坚持不懈地追求技术卓越和良好设计，这将提升敏捷能力。
- 要做到简单，即尽最大可能减少不必要的工作，这是一门艺术。
- 最佳的架构、需求和设计出自于自组织的团队。
- 团队要定期反省如何能够做到更有效，并相应地调整团队的行为。

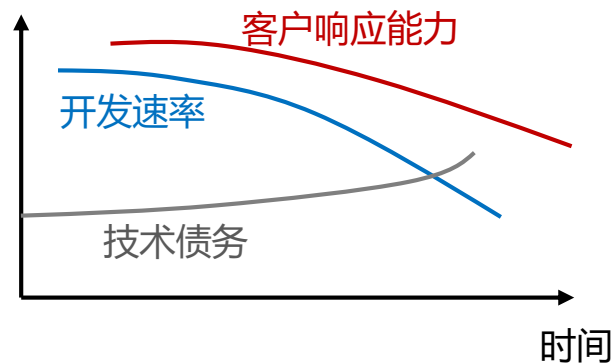
# 敏捷核心理念

## 聚焦客户价值

- 消除软件开发中的浪费
- 交付刚刚好的系统
- 随时构建质量，不容忍缺陷
- 及时消除技术债务，持续保持快速响应



软件业：45%的软件特性客户没有使用  
来源：Standish集团对5万个软件开发项目的调查

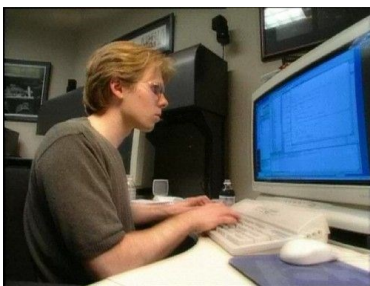


技术债务的积累对开发速率以及客户响应能力的影响

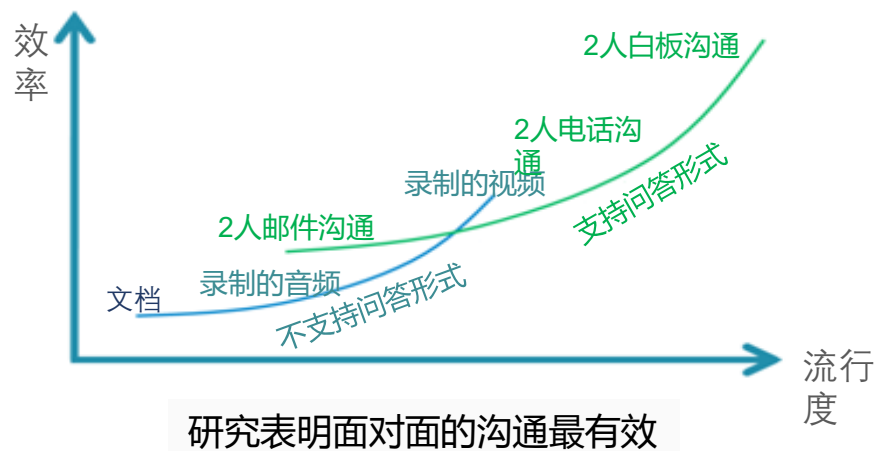
# 敏捷核心理念

## 激发团队潜能

- 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



研究表明：1981年来自不同公司的优秀程序员生产率之比是7:1，而2007年最新的研究数据则是40:1。

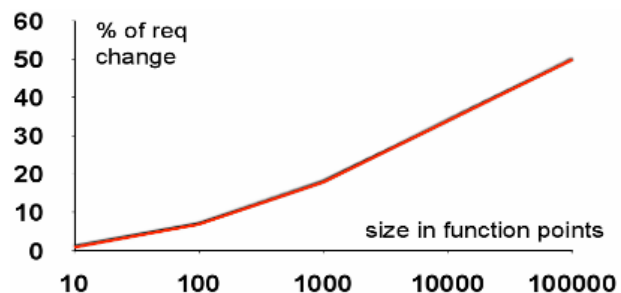
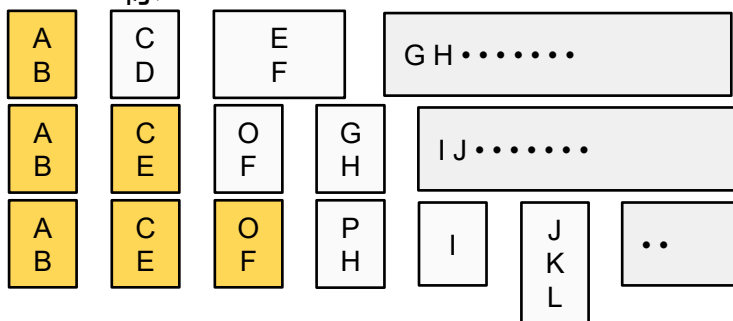


# 敏捷核心理念

## 不断调整以适应变化

- 客户是逐步发现真正需求
- 小批量是快速交付的关键
- 通过迭代计划不断调整以适应变化
- 应持续保持良好的软件架构
- 利用多层次反馈不断调整以逼近目标

标



随软件规模增长，需求变化呈非线性增长

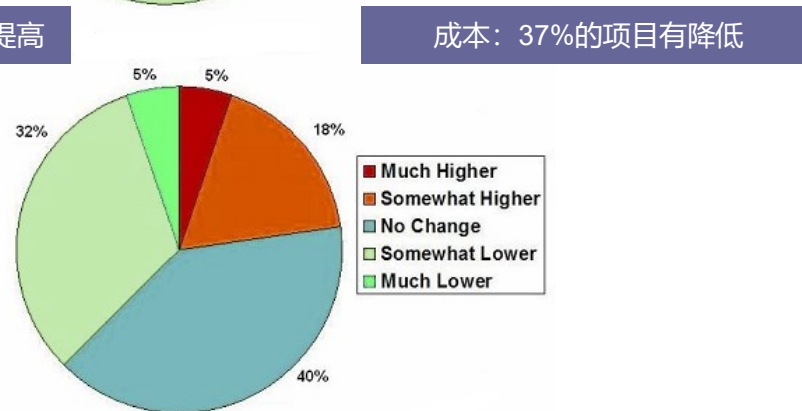
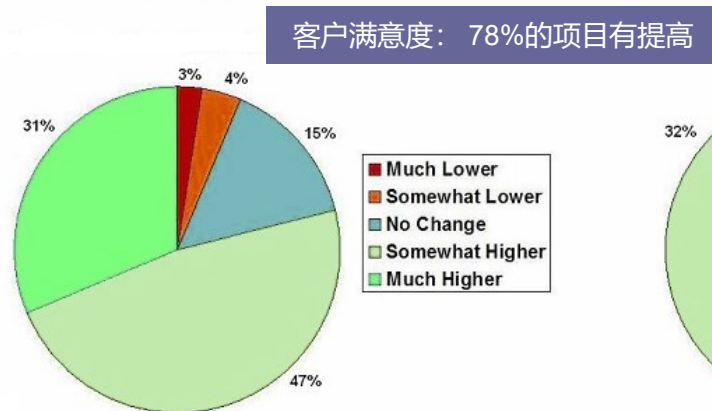
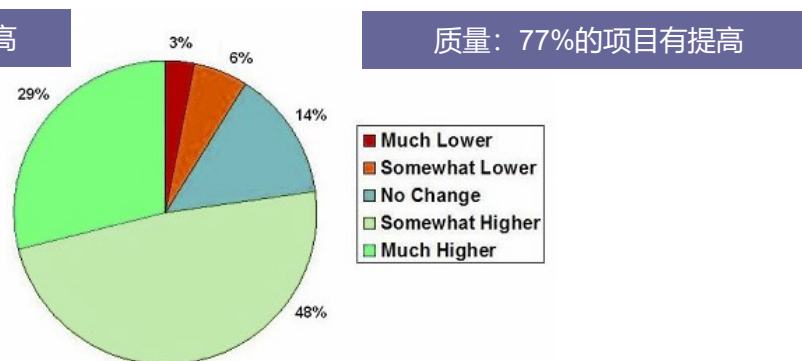
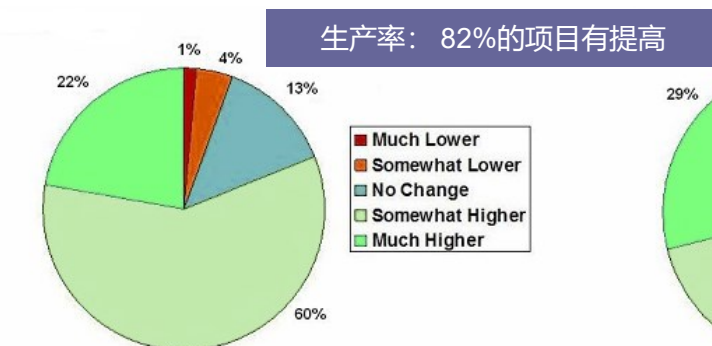
变化无法一次性预测，应根据迭代积累的经验和需求变化的情况，对计划进行不断地调整和细化。

# 敏捷开发应用



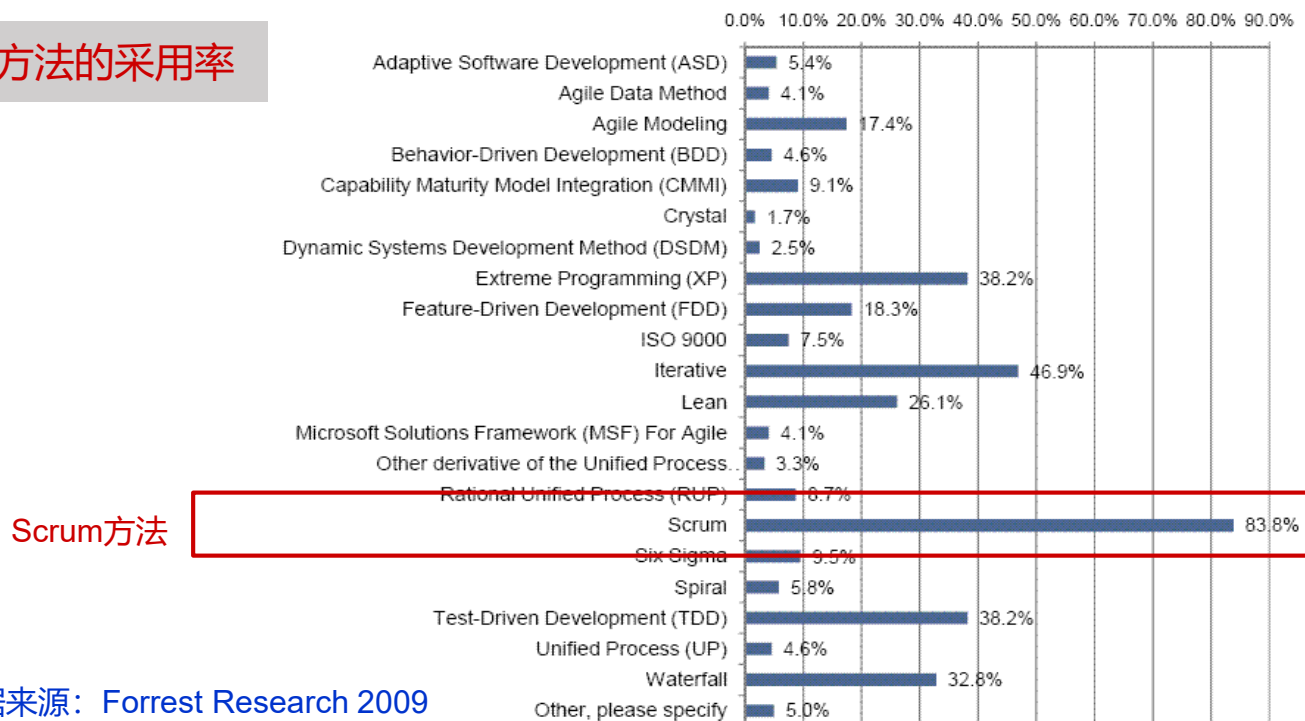
- ISO 9000 (09版) 标准将在原来八大原则的基础上新增敏捷原则
- 2000年美国军方软件开发标准 (DOD 5000.2) 推荐迭代为软件开发优选模式
- 2013年发布的新版PMBOK增加迭代及增量生命周期 (即对应敏捷模型)

# 敏捷开发应用



# 敏捷开发应用

## 不同开发方法的采用率



数据来源: Forrester Research 2009



# 教学提纲

2

## Scrum框架

- Scrum框架介绍
- Scrum团队角色
- Scrum制品和活动
- 敏捷规划与可视化管理

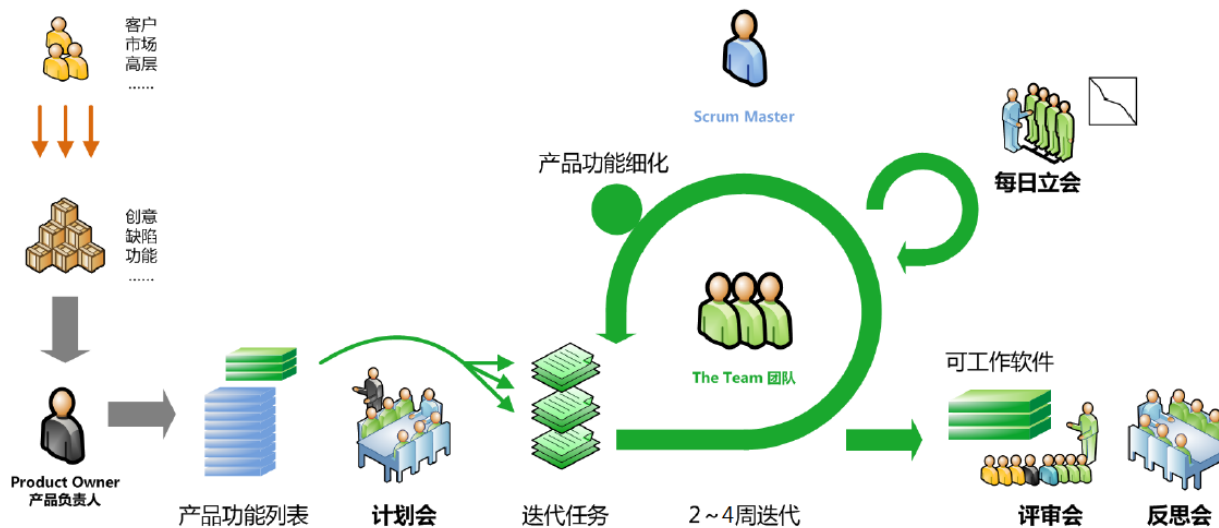
# Scrum方法

Scrum方法是1995年由Ken Schwaber和Jeff Sutherland博士共同提出，已被众多软件企业广泛使用，如Yahoo, Microsoft, Google, Motorola, SAP, IBM 等。



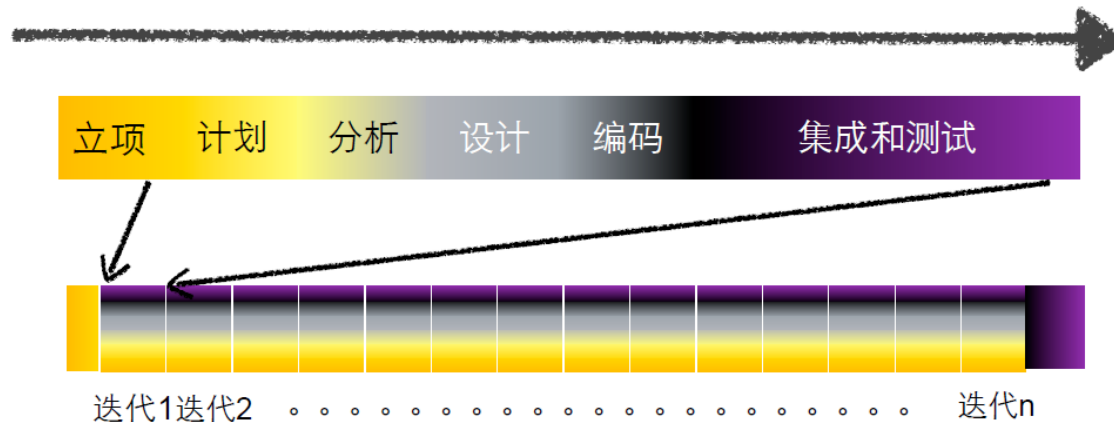
# Scrum框架

Scrum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，每个迭代周期成为一个冲刺（Sprint）。



# Scrum迭代开发

迭代开发将整个软件生命周期分成多个小的迭代（一般2~4周），每一次迭代就是一个小的瀑布模型，包括需求分析、设计、实现和测试等活动，结束时都要生成一个稳定和被验证过的软件版本。

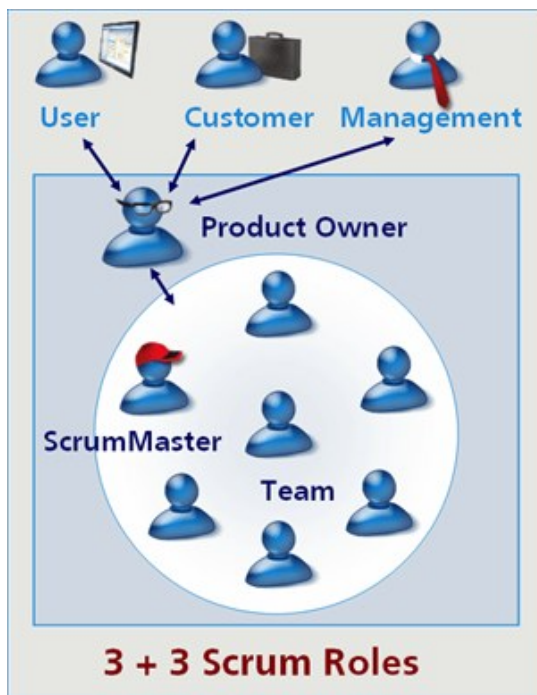


# Scrum迭代开发

## 迭代开发的关键要点：

- 每一次迭代都建立在稳定的质量基础上，并做为下一轮迭代的基线，整个系统的功能随着迭代稳定地增长和不断完善。
- 每次迭代要邀请用户代表验收，提供需求是否满足的反馈。
- 在一次迭代中，一旦团队作出承诺，就不允许变更交付件和交付日期；如果发生重大变化，产品负责人可以中止当次迭代。
- 在迭代中可能会出现“分解”和“澄清”，但是不允许添加新工作或者对现有的工作进行“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



## 产品负责人 (Product Owner)

职责：定义开发目标以及需要实现的特性和优先级。

## Scrum主管 (Scrum Master)

职责：保证团队高效而不受打扰地工作，优化工作条件和过程。

## 团队成员 (Team)

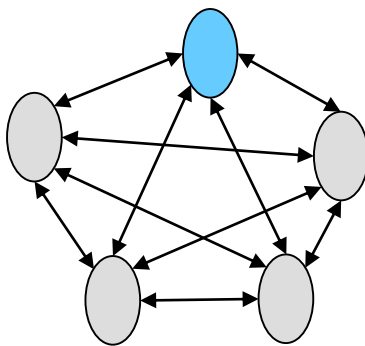
职责：自组织地完成项目开发，使用一切可行手段保证进度和质量。

# 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>保护团队不受打扰</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>

# Scrum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：同等的项目参与权激发大家的创造力，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



# Scrum团队组织

## 全功能的整体团队

- 一般 6~8 人
- 拥有多技能的、跨职能协作
- 关注于向项目干系人交付价值
- 自指导、自组织、可持续的速度

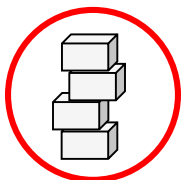


### 角色交叉

- 开发、测试、UI设计、文档编写等
- 基于技能而不是“岗位”认领工作

- 拥有共同目标，分担责任，彼此承诺致力于目标实现
- 拥有足够授权和资源，解决问题，找到自己的成功之路
- 有效的沟通和信息的透明

# Scrum制品



Product Backlog

产品订单是从客户价值角度理解的产品功能列表。

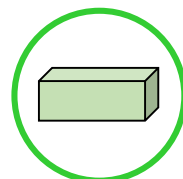
- 功能、缺陷、增强等都可以是产品订单项
- 整体上从客户价值进行优先级排序



Sprint Backlog

迭代订单是从开发技术角度理解的迭代开发任务。

- 简单环境：可直接把产品订单项分配到迭代中
- 复杂环境：可把一个产品订单项分为Web/后台.....软件/硬件.....程序/美工.....等开发任务



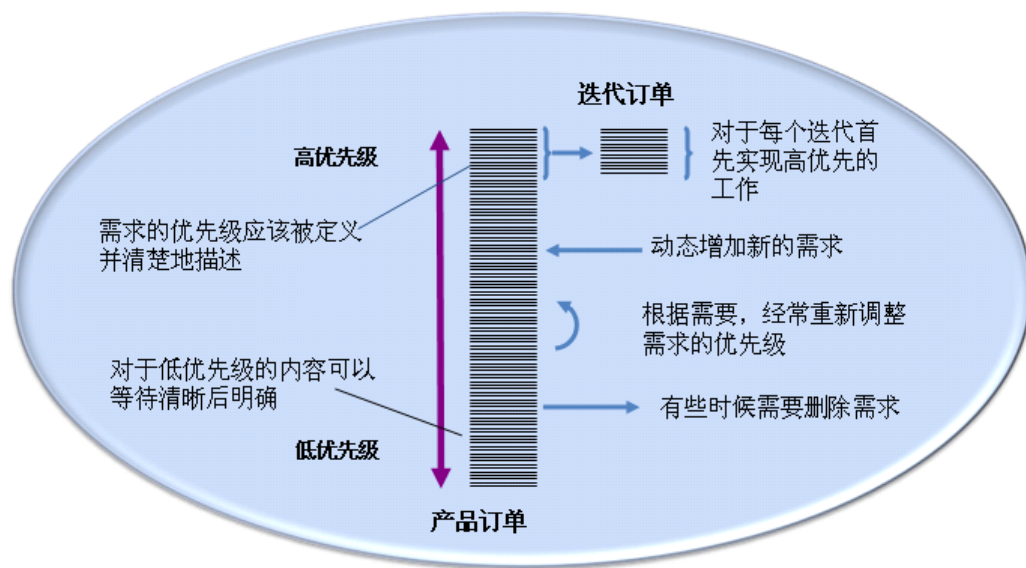
Working Software

可工作软件是可交付的软件产品。

- “可交付” 应视不同情况提前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。

# 产品订单

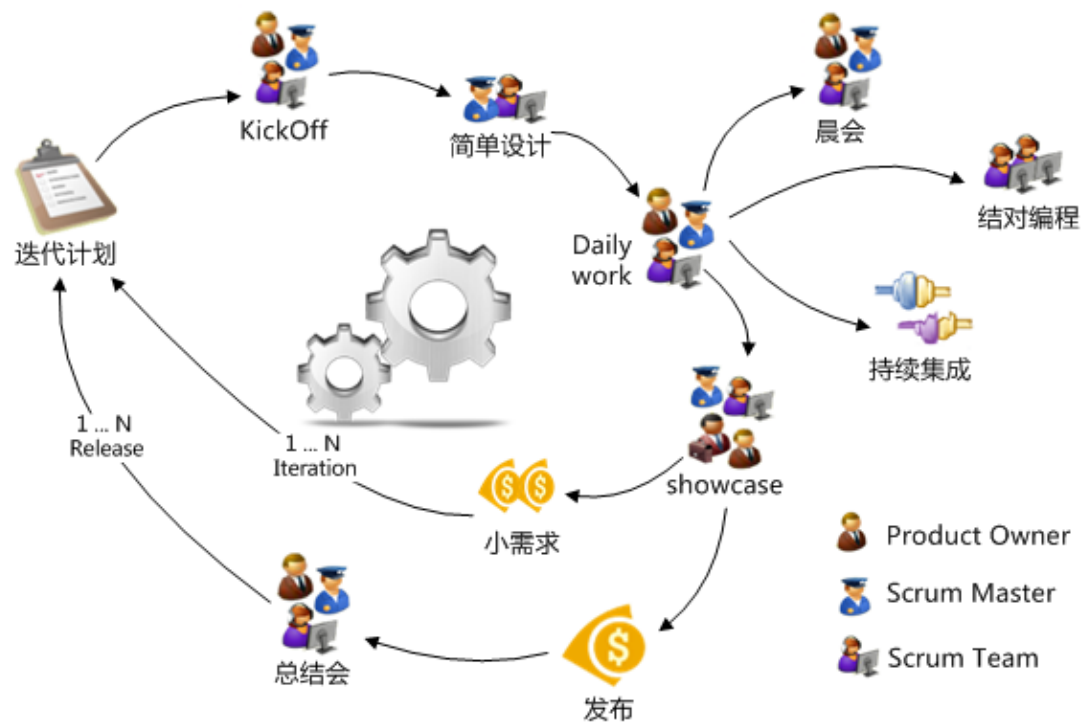
在迭代规划时，产品负责人告诉开发团队需要完成产品订单中的哪些订单项，开发团队决定在下一次迭代中他们能够承诺完成多少订单项。在迭代的过程中，没有人能够变更迭代订单，这意味着在一次迭代中需求是被冻结的。



# 产品订单

优先级	名称	用户故事描述
1	浏览商品	作为一名顾客想购买商品而不确定型号时，我希望能浏览网站在售的商品，按照①商品类型和②价格范围进行过滤。
2	搜索商品	作为一名顾客在查找某种商品时，我希望能进行不限格式的文本搜索，例如按照短语或关键字。
3	注册账户	作为一名新顾客，我希望注册并设置一个帐户，包括用户名、密码、信用卡和送货信息等。
4	维护购物车	作为一名顾客，我希望能将指定商品放入购物车（稍后购买）、查看我的购物车内的商品以及移除我不想要的物品。
5	结账	作为一名顾客，我希望能完成我购物车内所有商品的购买过程。
6	编辑商品规格	作为一名工作人员，我希望能够添加和编辑在售商品的详细信息（包括介绍、规格说明、价格等）。
7	查看订单	作为一名工作人员，我希望能登录并查看一段时间内应该完成或已经完成的所有订单。

# Scrum活动



# 迭代规划会议

**迭代规划会议**在每次迭代（或冲刺）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。

整个会议分成两部分：

- 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



# 迭代规划会议—第一部分

## 会议目的：

- 该会议的工作以分析为主，目的是要详细理解最终用户到底要什么，产品开发团队可以从该会议中详细了解最终用户的真实需要。
- 产品负责人在会前准备：条目化的需求（用户故事）及其优先级排序，最近1~2次迭代最希望看到的功能。会前准备至关重要，可帮助产品负责人理清头绪，不至于在迭代期内频繁提出变更、增加或删除故事。

## 基本要求：

- 经过估算和排序的产品订单列表
- 挂图、马克笔、剪刀、胶水、即时贴、白板、铅笔和蜡笔
- 假期计划表、重要人员的详细联系信息
- 参会成员：团队成员、Scrum主管、产品负责人

# 迭代规划会议—第一部分

## 会议过程：

- ①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。
- ②分析并明确用户验收测试。
- ③找出非功能性需求（性能、稳定性、... ..）。
- ④找出验收条件。
- ⑤弄清楚需要“完成”到何种水平。
- ⑥获得订单条目各个方面的清晰了解。
- ⑦绘制所需交付物的相关图表，包括流程图、UML图、手绘草图、屏幕界面设计等。
- ⑧回到步骤①，选取下一个产品订单条目。



# 迭代规划会议—第一部分

## 会议输出：

- 选择好的 Product Backlog 条目
- 各个 Backlog 条目的需求
- 各个 Backlog 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：

- 在 Sprint 规划会议第一部分结束前，团队需要短暂讨论一下，看看他们到底认为自己能完成多少工作。
- 询问团队能否快速回答下列问题，只需要简要回答即可：“我们能在这次迭代中完成第一个订单条目吗？”如果能得到肯定的回答，那么继续询问下一个条目，一直到已经分析完的最后一个 条目。

# 迭代规划会议—第二部分

## 会议过程：

- ①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。
- ②围绕该 Backlog 条目进行设计，并基于下列类似问题：
  - 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？
  - .....
- ③ 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。
- ④ 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。

# 迭代规划会议—第二部分

## 会议输出：

- 应用设计、架构设计图、相关图表
- 确保团队知道应该如何完成任务
- 注意：不要估算任务，不要分配任务



# 每日站立会议

## 会议目的：

- 团队在会议中做计划，协调其每日活动，还可以报告和讨论遇到的障碍。
- 任务板帮助团队聚焦于每日活动上，应在这个时候更新任务板和燃尽图。

## 基本要求：

- 任务板、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主管或管理层人员报告

# 每日站立会议

## 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。

## 每个团队成员需要回答三个问题：

- 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？



# 每日站立会议

## 会议输出：

- 团队彼此明确知道各自的工作，最新的工作进度图
- 最新的“障碍 Backlog”
- 最新的“Sprint Backlog”



# 迭代评审会议

## 会议目的:

- Scrum团队在会议中向最终用户展示工作成果，团队成员希望得到反馈，并以之创建或变更 Backlog 条目。

## 基本要求:

- 由团队展示有可能发布的产品增量
- 允许所有参与者（包括用户）尝试由团队展示的新功能
- 用户对团队演示的产品功能进行反馈

## 注意事项:

- 不要展示不可能发布的产品增量
- 团队不要针对产品负责人进行展示



# 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。

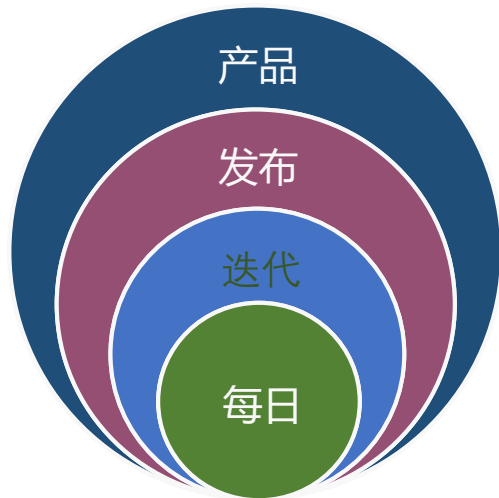
## 迭代回顾会议的关键要点：

- 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。





# Scrum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。

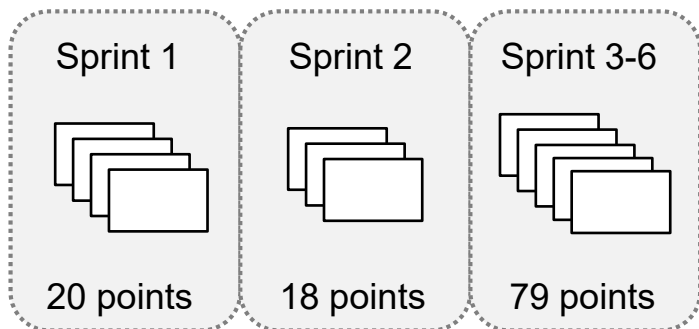
- **发布规划**是对整个产品发布过程的展望，其结果是产生产品订单。

- **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。

# Scrum规划

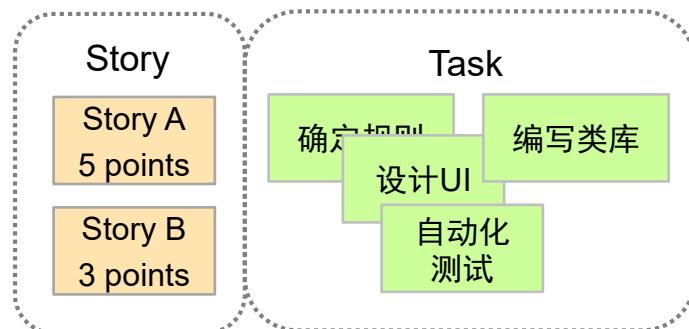
## 发布规划

- 定义用户故事并进行优先级划分
- 估算规模以及评估团队开发速度
- 制定发布计划



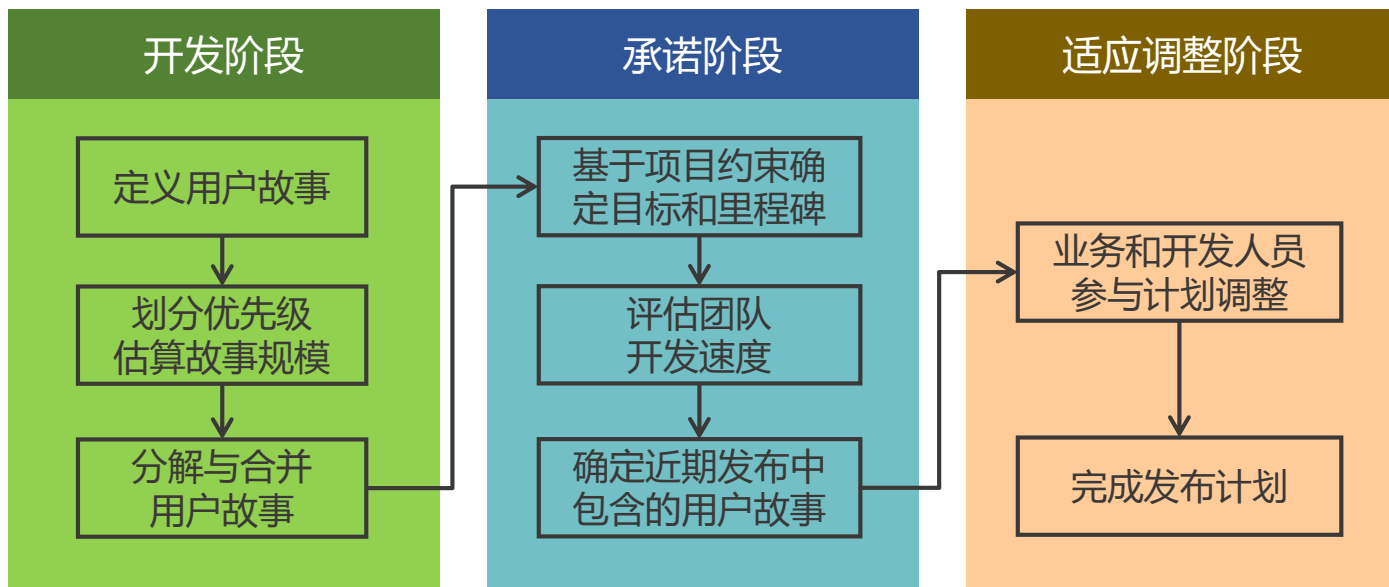
## 迭代规划

- 确定迭代目标并选择用户故事
- 将用户故事分解和细化到任务
- 对故事和任务进行时间估算



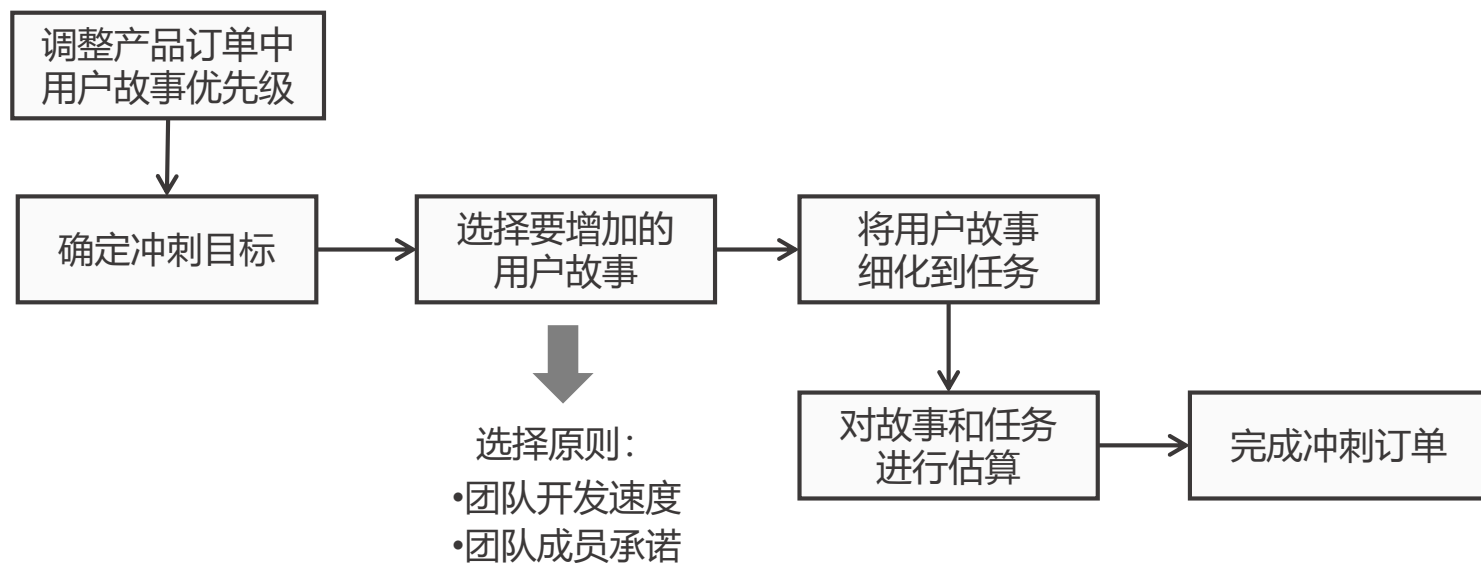
# Scrum规划

发布规划是对整个产品发布过程的展望，通常的规划周期是3~6个月。



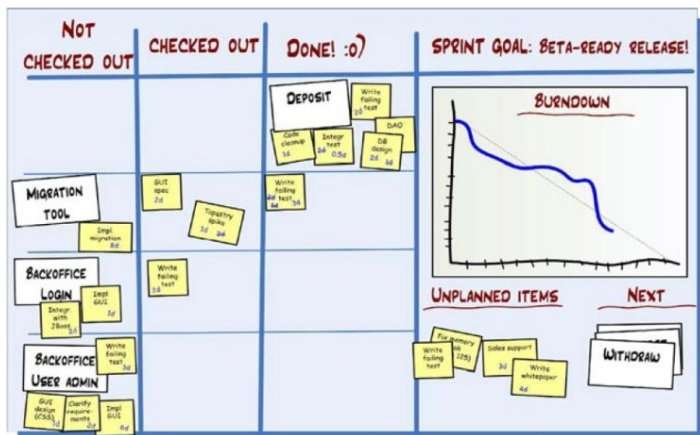
# Scrum规划

迭代规划将产生Scrum中提到的迭代订单，一般迭代周期为4周的时间。



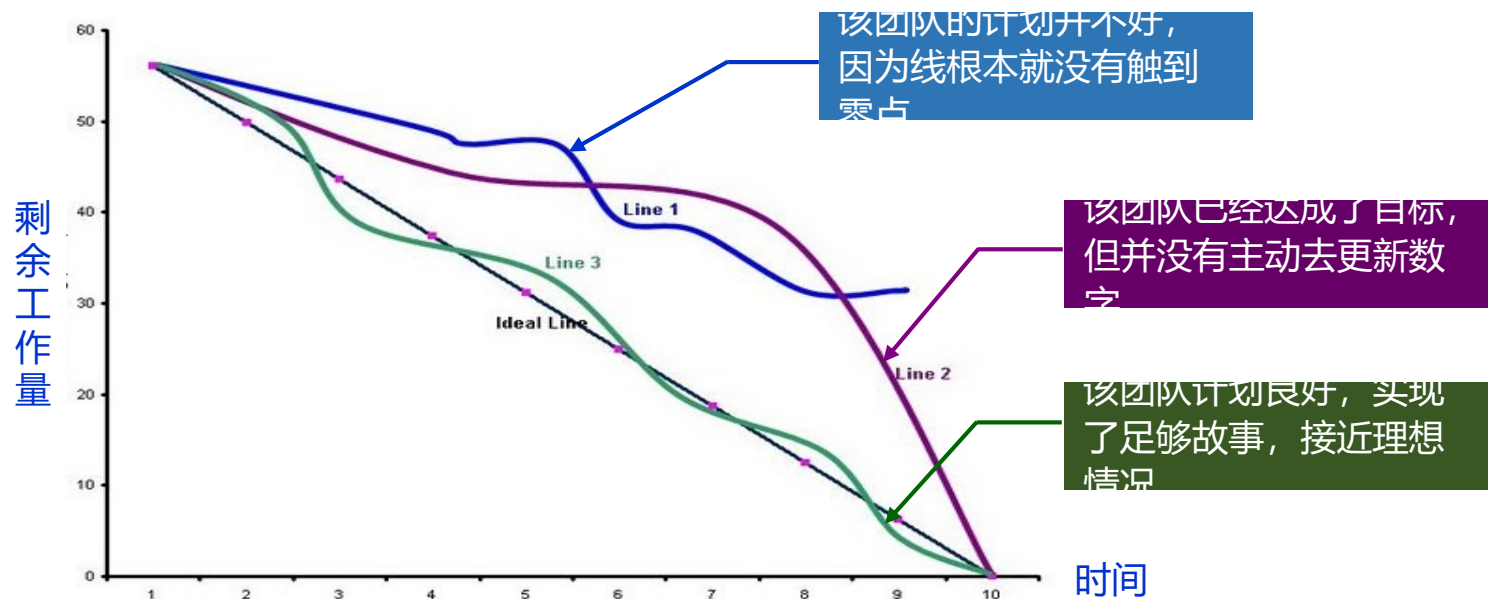
# 可视化管理

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。



# 可视化管理

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。



# 教学提纲

3

## 敏捷开发实践

- 用户故事及其估算
- 结对编程
- 测试驱动开发
- 持续集成

# 用户故事

用户故事 (User Story) 是从用户角度对功能的简要描述。

格式：作为一个<角色>，可以<活动>，以便于<价值>。

谁要使用这个功能?

需要执行什么操作?

完成操作后带来什么好处?



# 用户故事

优先级	名称	用户故事描述
1	浏览商品	作为一名顾客想购买商品而不确定型号时，我希望能浏览网站在售的商品，按照①商品类型和②价格范围进行过滤。
2	搜索商品	作为一名顾客在查找某种商品时，我希望能进行不限格式的文本搜索，例如按照短语或关键字。
3	注册账户	作为一名新顾客，我希望注册并设置一个帐户，包括用户名、密码、信用卡和送货信息等。
4	维护购物车	作为一名顾客，我希望能将指定商品放入购物车（稍后购买）、查看我的购物车内的商品以及移除我不想要的物品。
5	结账	作为一名顾客，我希望能完成我购物车内所有商品的购买过程。
6	编辑商品规格	作为一名工作人员，我希望能够添加和编辑在售商品的详细信息（包括介绍、规格说明、价格等）。
7	查看订单	作为一名工作人员，我希望能登录并查看一段时间内应该完成或已经完成的所有订单。

# 敏捷估算

**故事点的基本做法：**把一些常见的“标准任务”给出一个“标准点数”，形成比较基线；估算时只要是同一类型任务，直接写故事点数而非天数。

- 故事点是用于表达用户故事、功能或其他工作的总体规模的度量单位，它是一个相对度量单位。
- 理想日是用于表达用户故事、功能或其他工作的总体规模的另外一种度量单位，它是一个绝对度量单位。
- 理想时间是某件事在剔除所有外围活动以后所需的时间；一般为一天有效工作时间的 60-80% 比较合理，但绝不会是全部。

# 敏捷估算

优先级	名称	用户故事描述	故事点
1	浏览商品	作为一名顾客想购买商品而不确定型号时，我希望能浏览网站在售的商品，按照①商品类型和②价格范围进行过滤。	10
2	搜索商品	作为一名顾客在查找某种商品时，我希望能进行不限格式的文本搜索，例如按照短语或关键字。	15
3	注册账户	作为一名新顾客，我希望注册并设置一个帐户，包括用户名、密码、信用卡和送货信息等。	1
4	维护购物车	作为一名顾客，我希望能将指定商品放入购物车（稍后购买）、查看我的购物车内的商品以及移除我不想要的物品。	40
5	结账	作为一名顾客，我希望能完成我购物车内所有商品的购买过程。	28
6	编辑商品规格	作为一名工作人员，我希望能够添加和编辑在售商品的详细信息（包括介绍、规格说明、价格等）。	12
7	查看订单	作为一名工作人员，我希望能登录并查看一段时间内应该完成或已经完成的所有订单。	7

# 敏捷估算

敏捷估算扑克本质上是扑克牌，它基于Delphi估算原理，可以快速地估算出需要的数字。



- 估算扑克牌上的数字：有些牌是自然数排列，有些是斐波纳契数，有些则是不连续自然数。
- 具体选用哪种扑克，要根据被估算的内容的跨度大小而定，如果估算值跨度在10倍以内，那么采用顺序自然数比较好，如果数值跨度较大，达到10倍以上，那么采用斐波纳契数比较好。
- 一般而言，估算软件开发工时的话，自然数可能更好一些，毕竟数值都不大，跨度也不会很夸张。

# 敏捷估算

**分牌：**每名参与估算的成员分得相同花色的一组牌，两张Joker不参与估算。



敏捷扑克和普通游戏扑克一样，也有54张牌，也拥有4种花色（每种各13张）和两张Joker。

敏捷扑克的每种花色均是一组13张牌组成的估算扑克牌，牌正面上印刷有供估算用的数字与符号，数字分别是1/2、1~10和20以及符号“!”（代表一些未知情况，如无法提供准确估算值等）。

- 一副估算扑克可供四人使用，如果参与的人员多于4人，可使用多副扑克。
- 一般推荐4-8人参与估算，人太少会使估算结果偏差很大，而太多会拉长估算时间，降低估算效率。

# 敏捷估算

**讲解订单故事：**产品负责人从Backlog中选择一个条目，为大家详细讲解该条目；团队成员进行讨论并提问，产品负责人逐一解答大家的问题。



在讲解过程中，千万不要制定该条目的负责人或有明显倾向的人来做这个条目，这样会大大降低团队成员的积极性，甚至会扰乱估算秩序与结果。

这个步骤是团队和产品负责人之间的交互环节，帮助团队和产品负责人共同加深对条目的理解；产品负责人也会根据大家的反馈，及时修改和完善条目。



# 敏捷估算

**估算：**当团队成员确认已经对该条目完全了解且无任何重大问题时，大家开始对该条目进行估算，同时选出代表自己估算值的纸牌，但不可立即亮牌。在估算过程中，为避免干扰估算结果，团队成员之间不可以互相商讨。当所有成员选牌完毕，大家可以同时亮牌。





# 敏捷估算



VS



**争论与讨论：**对比每张牌估算值之间的大小，若估算值差距明显，代表大家对该条目的价值没有获得共识，团队需要对该条目价值评估结果进行讨论。

**共识：**对该条目重新进行估算，直到团队的评估数值达成一致。一般情况下，最多三轮就可以得出一个比较统一的意见；如果三轮之后依然没有得到统一的意见，那么Scrum Master应当立即中断该条目的估算，取平均值或其他大家能接受的值作为估算结果。





# 简单设计

敏捷方法认为设计非常重要，因此应该是一个持续的事务。设计只考虑当前的需求定义，先尝试使用能够工作的最简单设计，然后随着现实的不不断显现来更改它。

- 第一次迭代搭建基本的系统框架
- 以后的迭代过程是在反馈和编程的基础上做交互式设计，减少了设计的投机性
- 重构对设计进行优化

对简单设计的需求并不是说所有设计都很小，也不表示它们是无足轻重的；它们只不过需要尽可能地简单，但是仍能工作。

# 结对编程

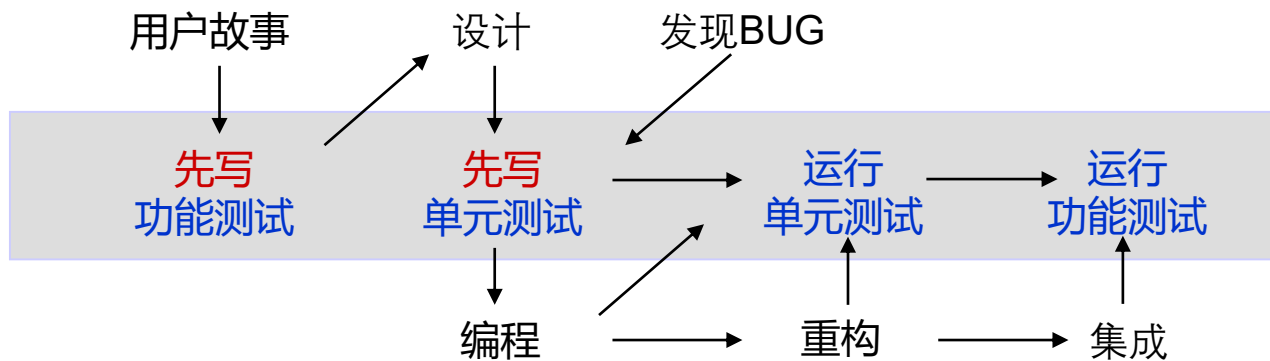
结对编程是由两名程序员在同一台电脑上结对编写解决同一问题的代码。



- 有助于提升代码设计质量;
- 研究表明结对生产率比两个单人总和低15%但缺陷数少15%，考虑修改缺陷工作量和时间比初始编程大几倍，所以结对编程总体效率更高；
- 能够大幅促进团队能力提升和知识传播；
- 实施时需要团队成员克服个性冲突和习惯差异。

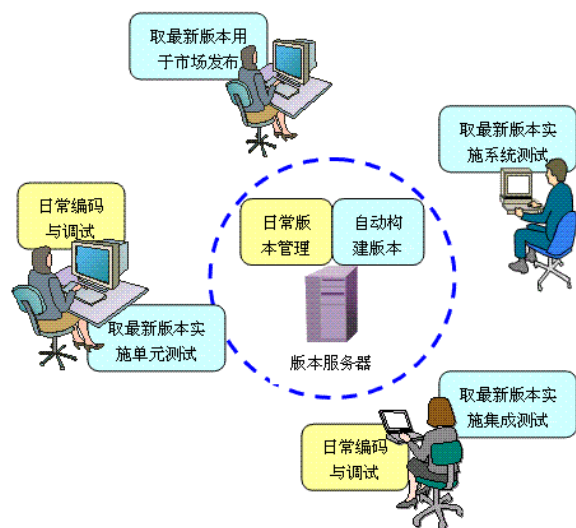
# 测试驱动开发

测试驱动开发是以测试作为编程中心，要求在编写任何代码之前，首先编写定义代码功能的测试用例，编写的代码要通过测试，并不断进行重构优化。



# 持续集成

持续集成是一项软件开发实践，团队成员经常集成自己的工作，通常每人每天至少集成一次，每次集成通过自动化构建完成。



- 所有开发人员需要在本地机器上进行本地构建，然后再提交到版本控制库中，以免影响持续集成。
- 开发人员每天至少向版本控制库中提交一次代码，至少从版本控制库中更新一次代码到本地机器。
- 需要有专门的集成服务器来执行集成构建，并通过自动化的构建（包括编译、发布、自动化测试）来验证，从而尽快地发现集成错误。

# Reference

- 清华大学国家级精品课程 《软件工程》 主讲人 刘强 副教授 刘璘 副教授
- [https://www.icourses.cn/sCourse/course\\_3016.html](https://www.icourses.cn/sCourse/course_3016.html)
- [https://www.xuetangx.com/course/THU08091000367/5883555?channel=learn\\_title](https://www.xuetangx.com/course/THU08091000367/5883555?channel=learn_title)



谢谢大家！

---

THANKS

