敏捷开发方法

清华大学软件学院 刘强

教学提纲

1. 敏捷方法概述
2. Scrum □□
3. 敏捷开发实践
4. 实验项目团队活动

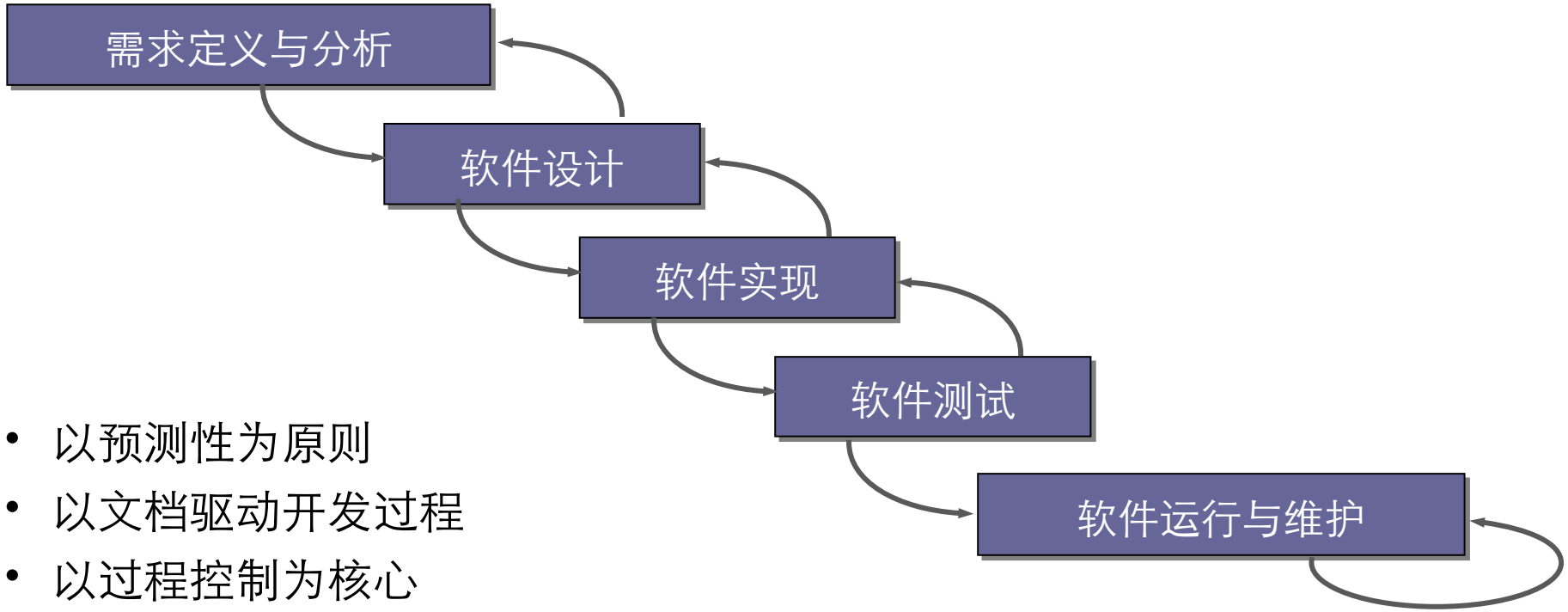
教学提纲

1 □□□□□□

- 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。

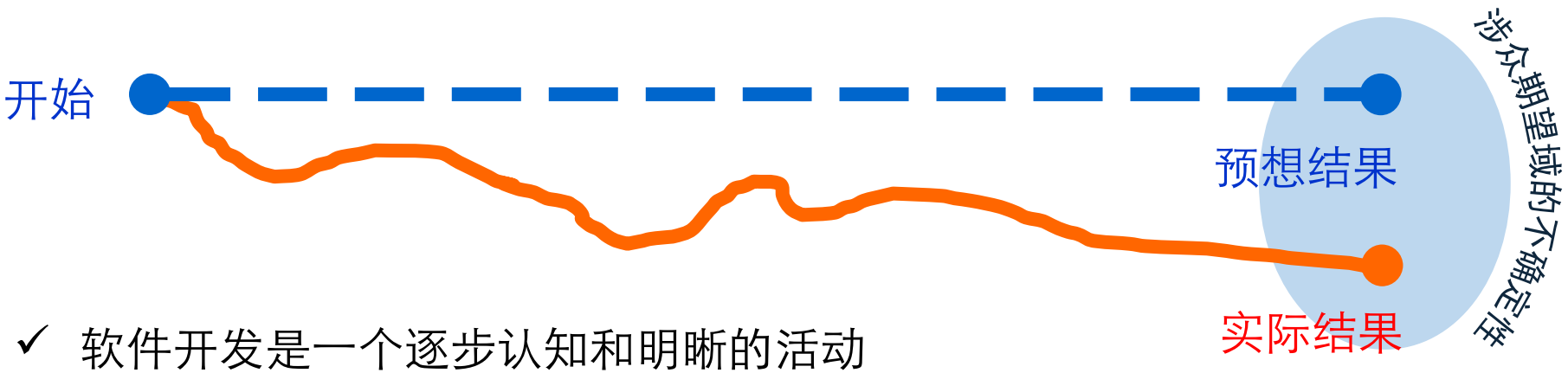


- 以预测性为原则
- 以文档驱动开发过程
- 以过程控制为核心

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

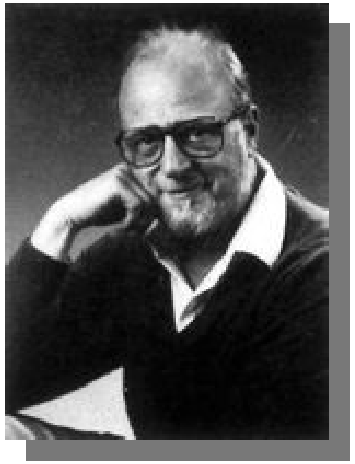
软件开发之道

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。

——— 杰拉尔德·温伯格

正确的软件

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□。

软件运行正确

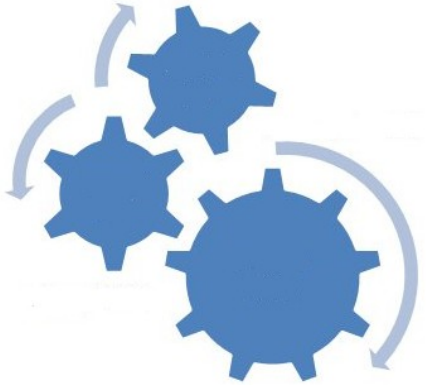
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

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

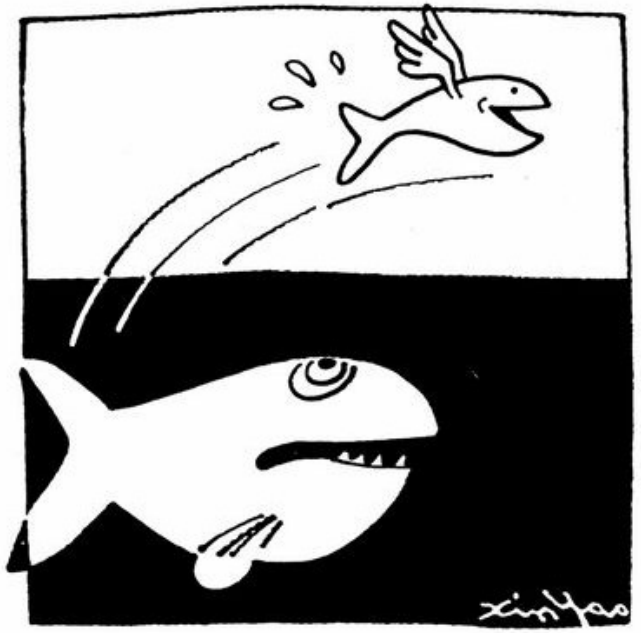
- ✓ 高质量的交付物是最重要的
- ✓ 系统不是一次构建而成，而是迭代演进的
- ✓ 基于完整的场景构建计划，并按优先级执行



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



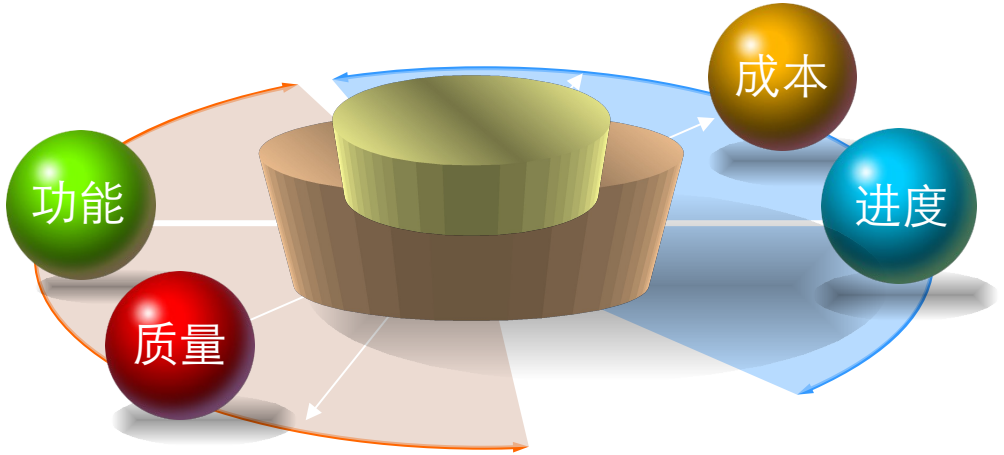
如今不是大鱼吃小鱼，而是快鱼吃慢鱼。

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。

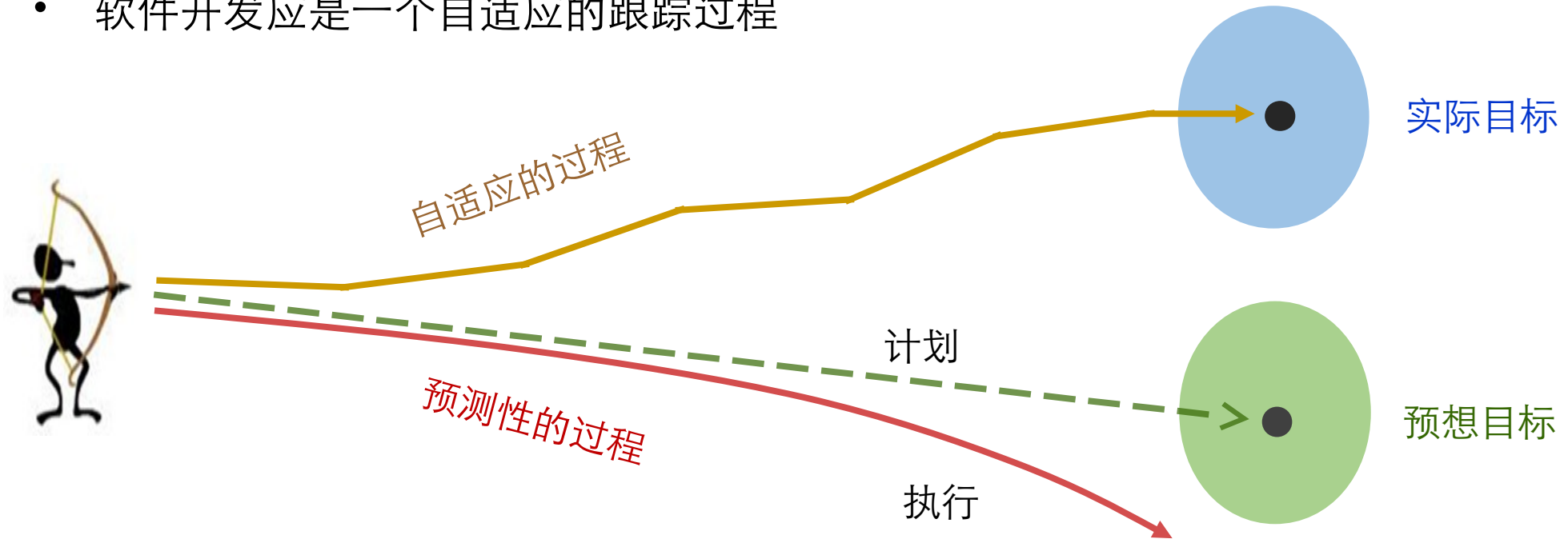
- 更关注协作
- 更关注质量
- 更关注可工作的产品
- 更关注全才化的专才
- 基于实践而非基于理论



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

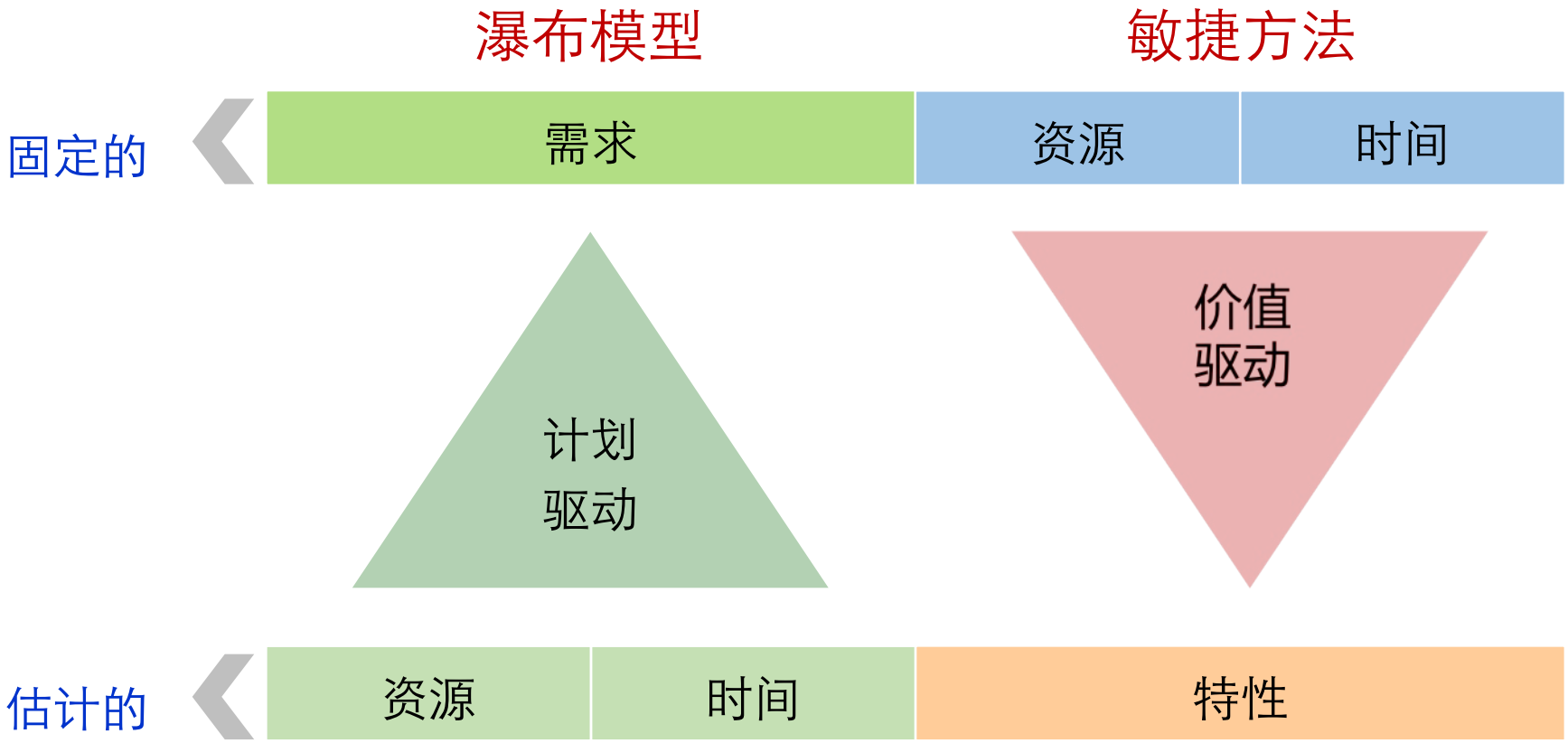
项目失败的原因最终都可追溯到某个信息没有及时准确地传递到应该接收它的人。

—— *Alistir 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。

—— *Alistir Cockburn*

敏捷开发方法



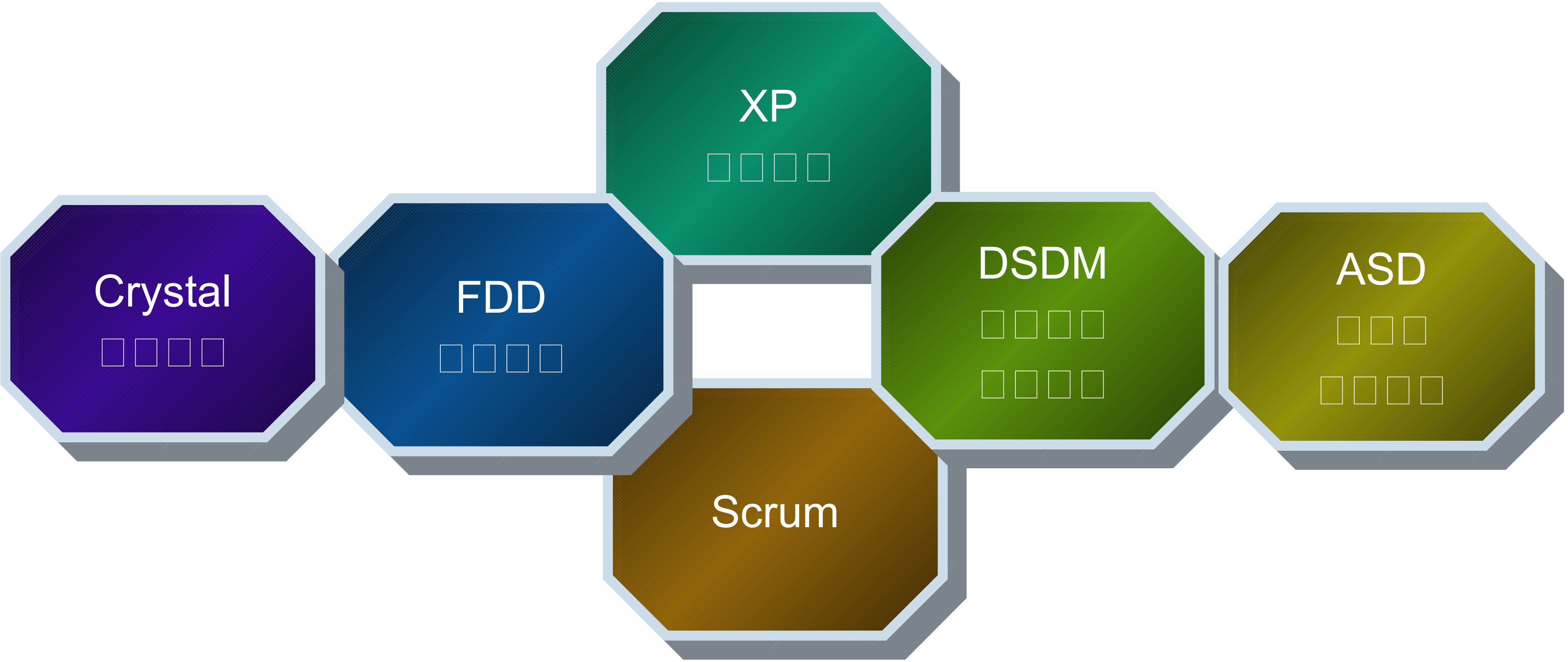
敏捷开发方法

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

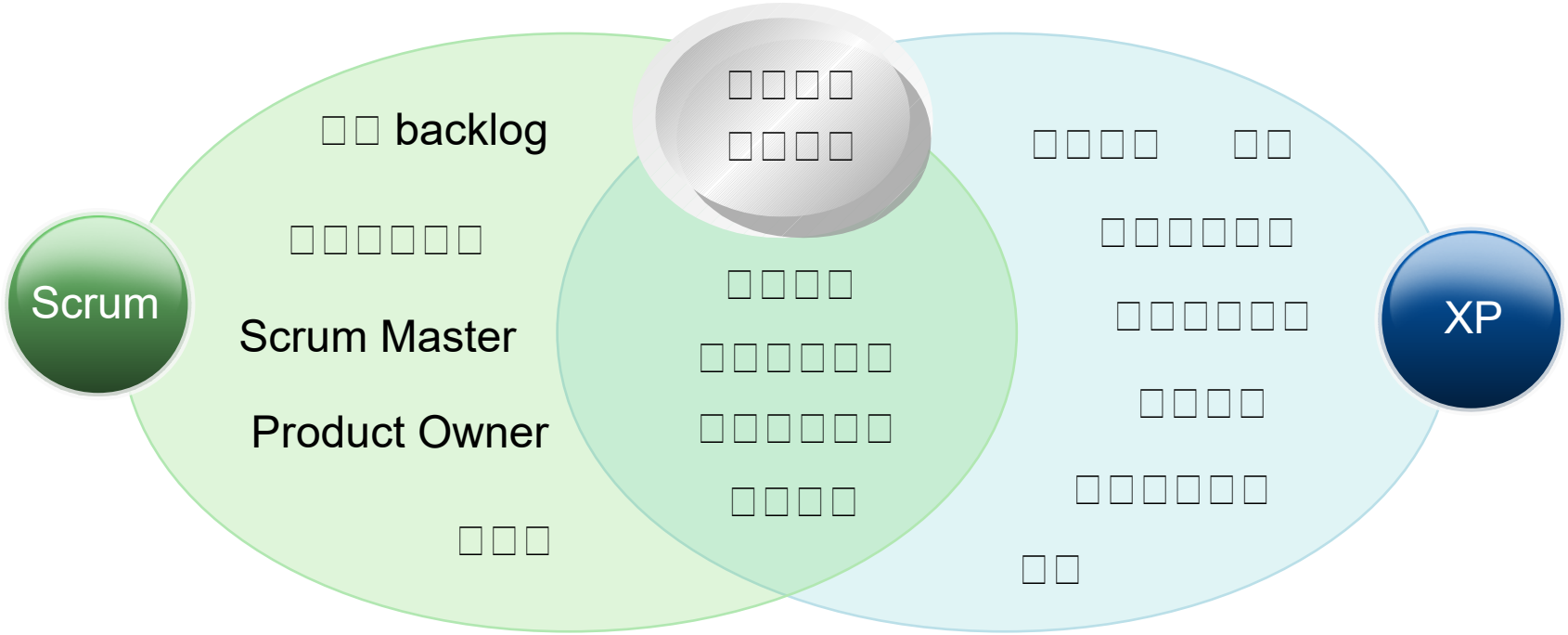
敏捷开发方法

敏捷开发方法



□□□
□□□ XP □□ Scrum □□□□□

敏捷开发方法



Scrum □□□□□□

XP □□□□□□

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。

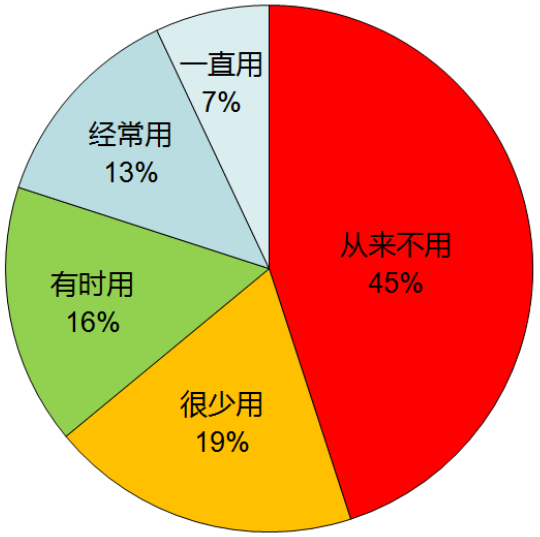
敏捷宣言

- [illegible]
- [illegible]——[illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]。

敏捷核心理念

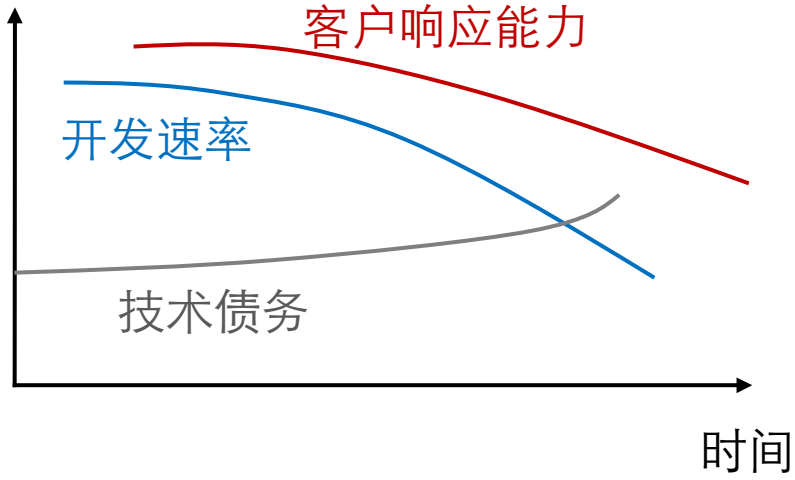
聚焦客户价值

- 消除软件开发中的浪费
- 交付刚刚好的系统



□□□□ 45% □□□□□□□□□□□□
□□□ *Standish* □□□ *5* □□□□□□□□□□□□

- 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



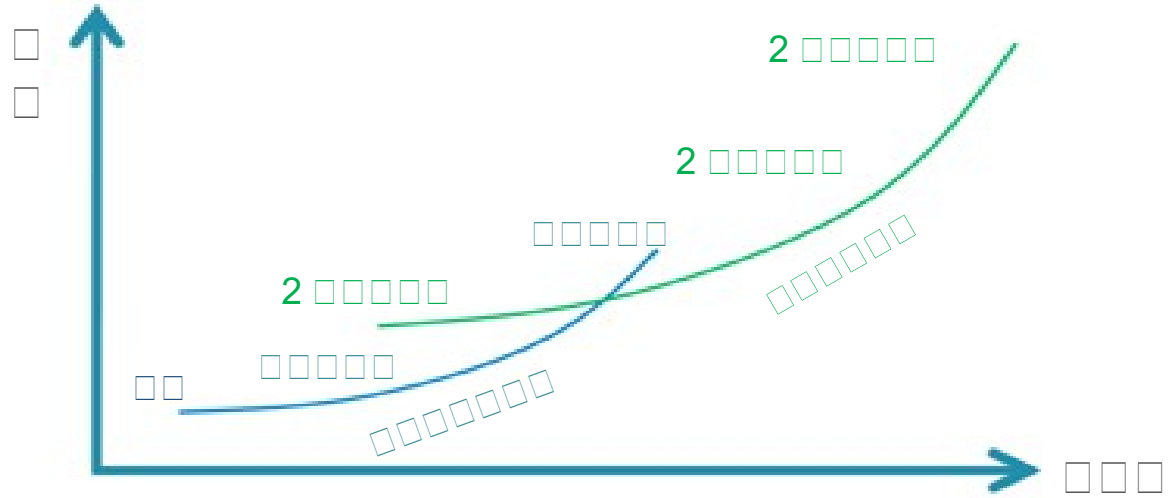
技术债务的积累对开发速率以及客户响应能力的影响

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

□□□□□ 1981 □□□□□□□□□□□□□□□□□□□ 7:1 □□ 2007 □□□□□□□□□□□ 40:1 □

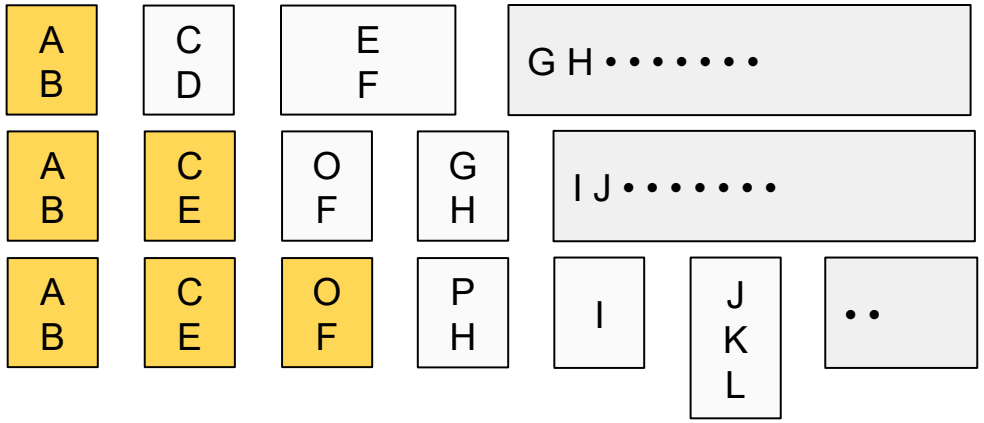


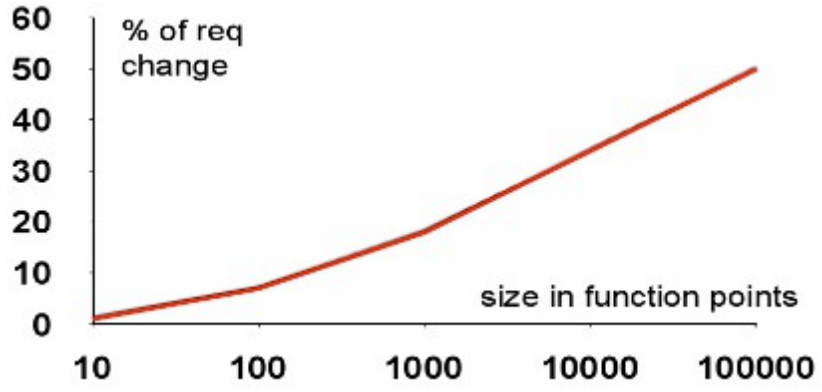
研究表明面对面的沟通最有效

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





随软件规模增长，需求变化呈非线性增长

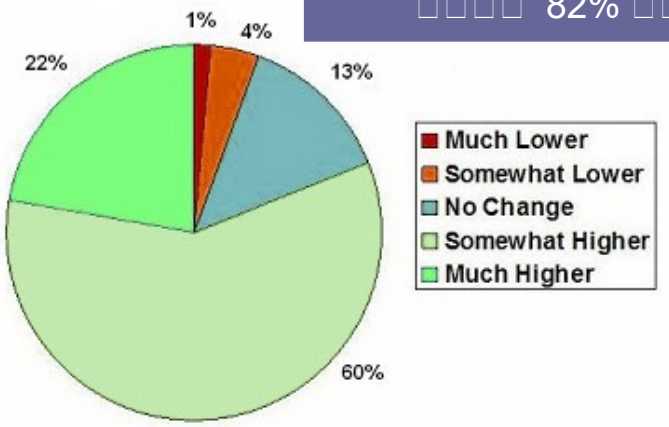
变化无法一次性预测，应根据迭代积累的经验和需求变化的情况，对计划进行不断地调整和细化。

敏捷开发应用

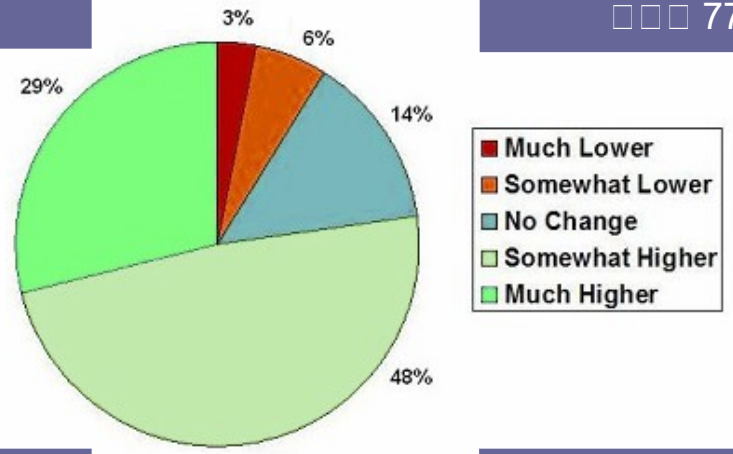
- ISO 9000 □ 09 □□□□□□□□□□□□□□□□□□□□□□
- 2000 □□□□□□□□□□□□ DOD 5000.2 □□□□□□□□□□□□□□
- 2013 □□□□□□ PMBOK □□□□□□□□□□□□□□□□□□□□□

敏捷开发应用

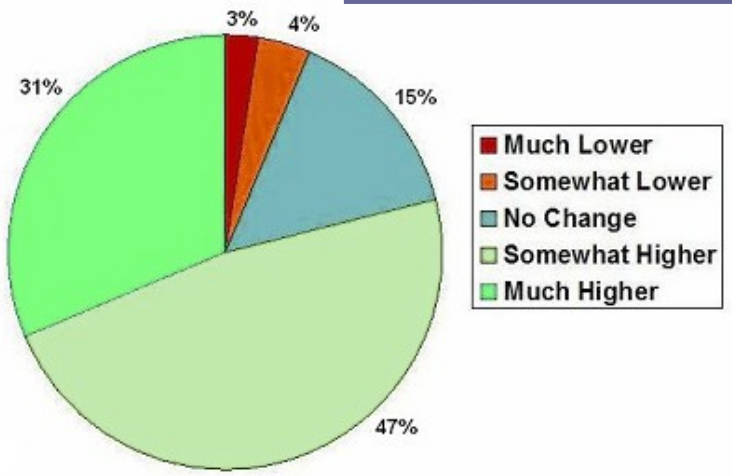
□□□□ 82% □□□□□□



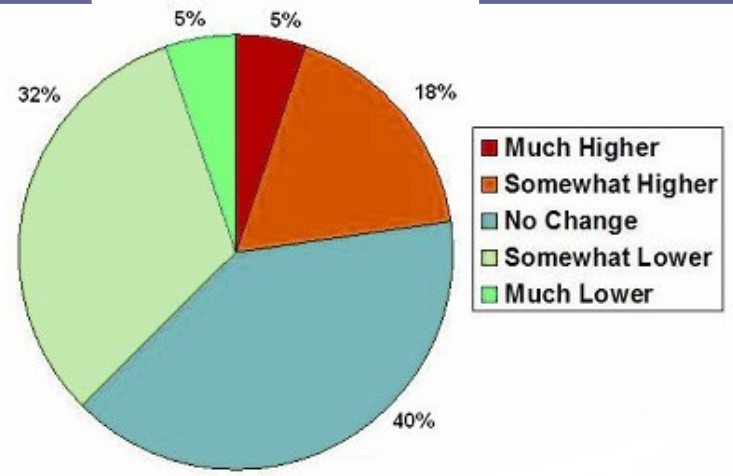
□□□ 77% □□□□□□



□□□□□□ 78% □□□□□□

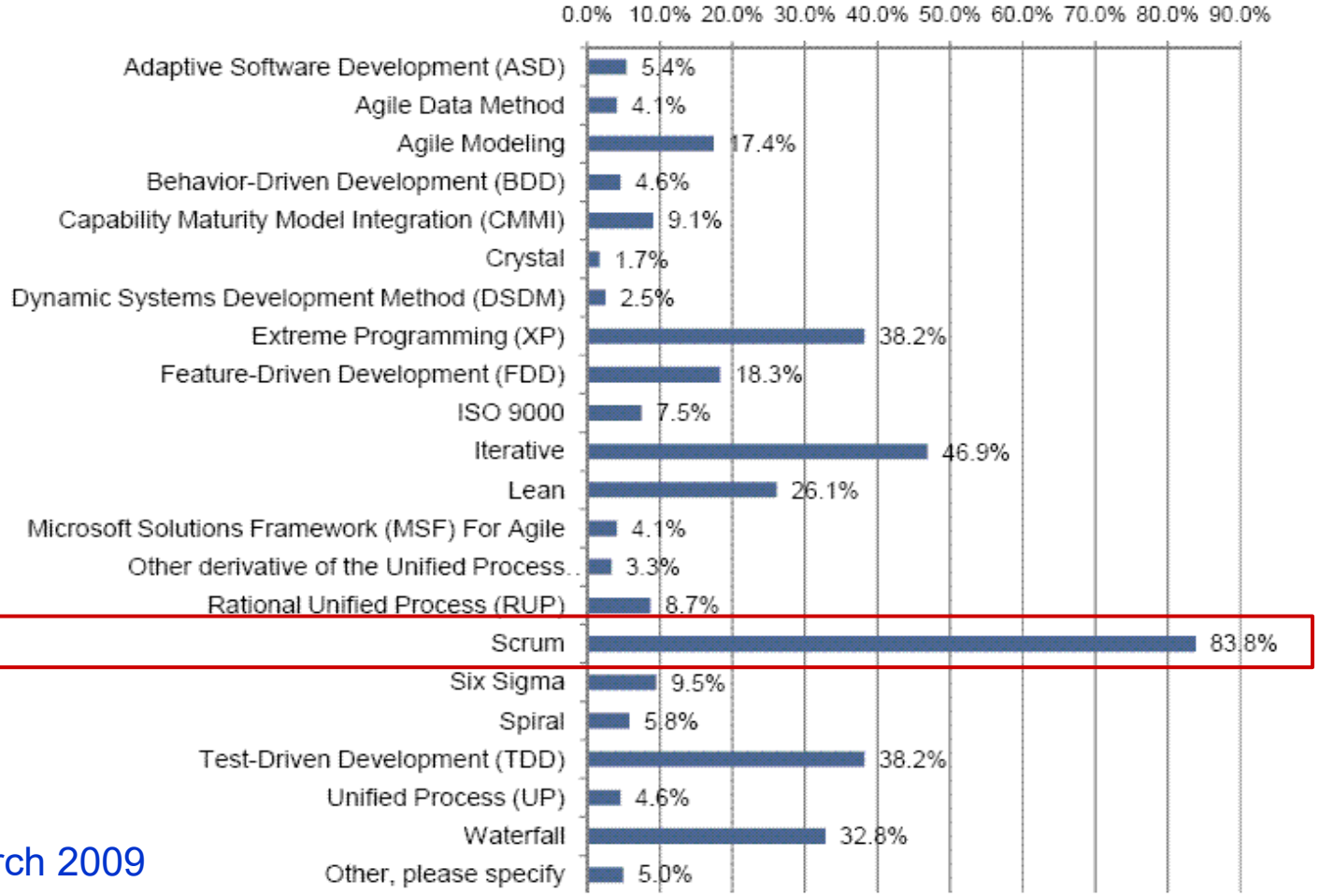


□□□ 37% □□□□□□



敏捷开发应用

□□□□□□□□□□



Scrum □□

□□□□□ Forrest Research 2009

教学提纲

2 Scrum □□

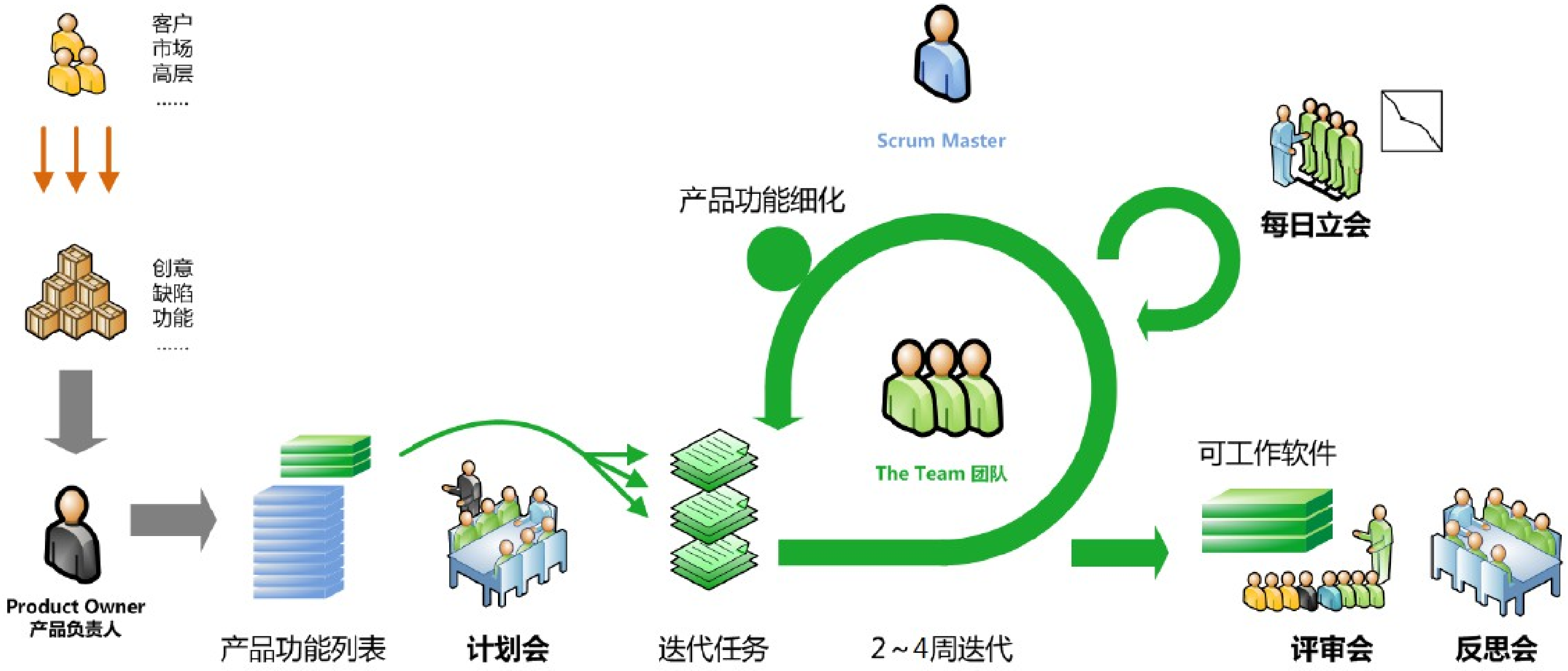
- Scrum □□□□
- Scrum □□□□
- Scrum □□□□□
- □□□□□□□□□□

Scrum □□

Scrum □□□ 1995 □□ Ken Schwaber □ Jeff Sutherland □□□□□□□□□□□□□□□□□□□□ Yahoo, Microsoft, Google, Motorola, SAP, IBM □□

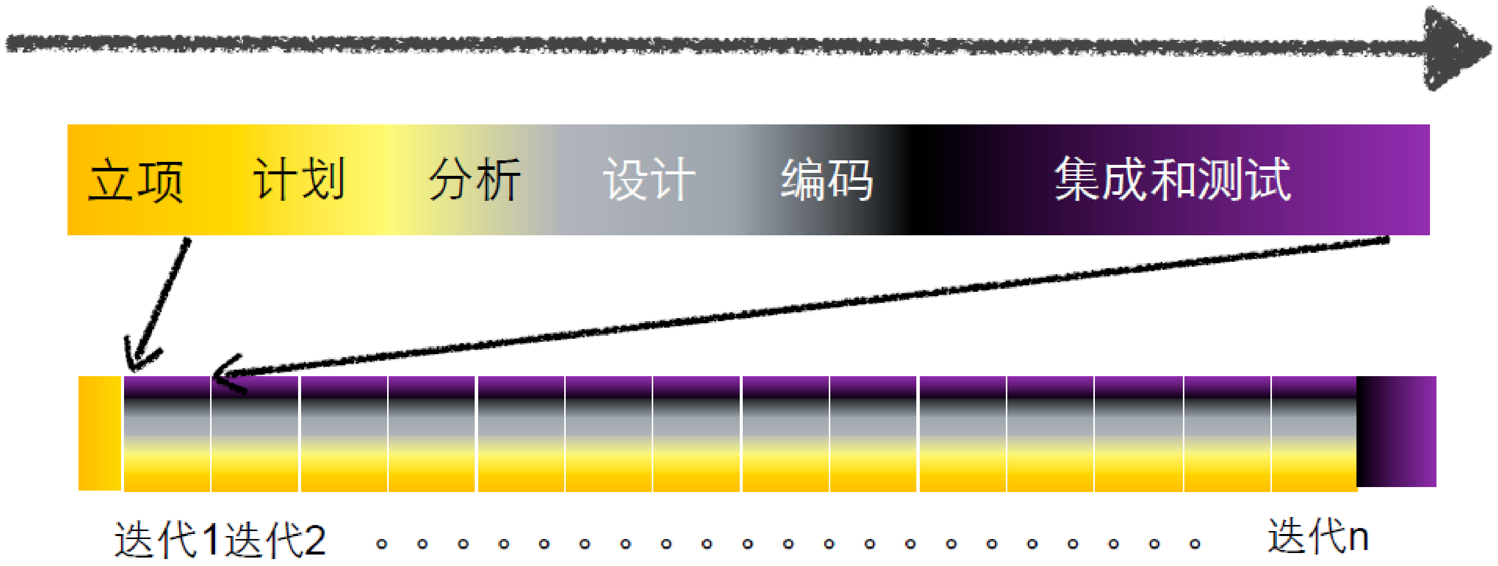
Scrum □□

Scrum □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□ Sprint □□



Scrum □□□□

□□□□□□□□□□□□□□□□□□□□□□□□2~4□□□□□□□□□□□□□□□□□
□□□

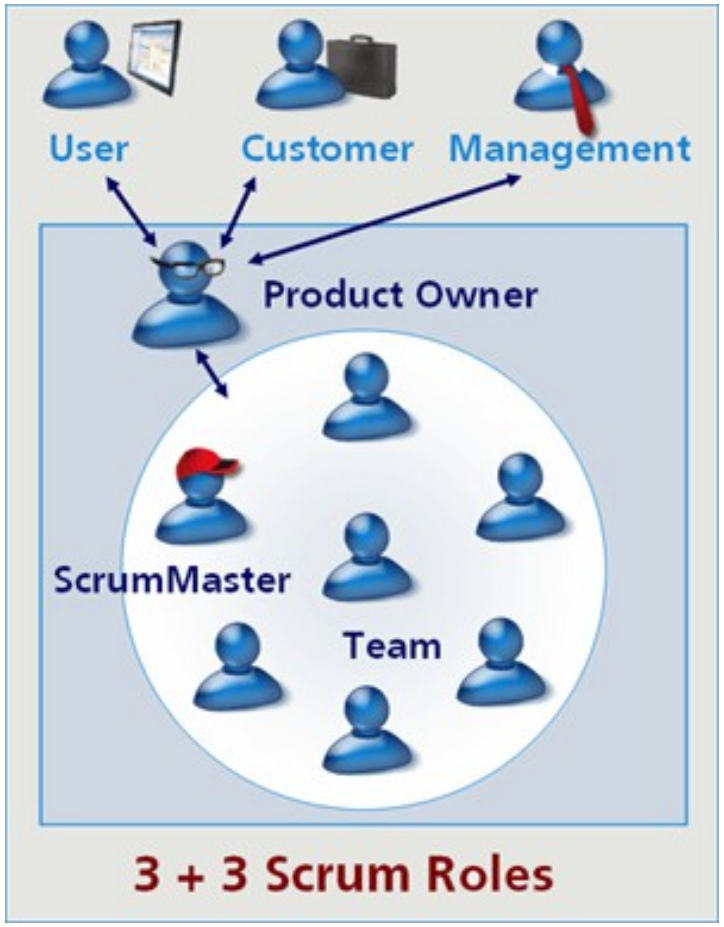


Scrum □□□□

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 □□□□



□□□□□□ Product Owner □

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Scrum □□□ Scrum Master □

□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□

□□□□□ Team □

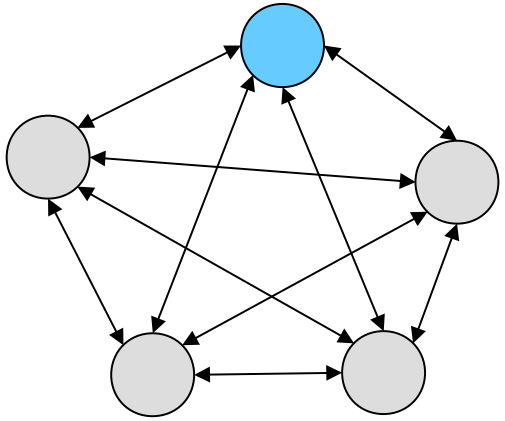
□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□

Scrum [illegible]

[illegible]	[illegible]	[illegible]
[illegible]	• [illegible] • [illegible] • [illegible] • [illegible]	[illegible]
Scrum [illegible]	• [illegible] • [illegible] • [illegible] • [illegible] • [illegible]	[illegible]
[illegible]	• [illegible] • [illegible] • [illegible]	• [illegible] 5-9 [illegible] • [illegible] • [illegible] • [illegible]

Scrum □□□□

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 □□□□

全功能的整体团队

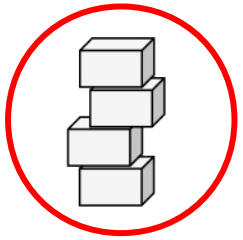
- □□ 6 □ 8 □
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

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 □□

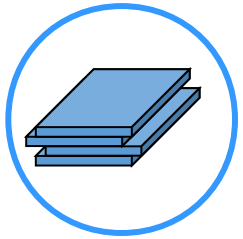


Product Backlog

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

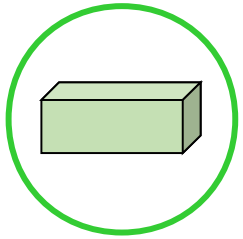


Sprint Backlog

□□□□□□□□□□□□□□□□□□□□□□□□

•□□□□□□□□□□□□□□□□□□□□□□

•□□□□□□□□□□□□□□□□□□ Web/ □□……□□ / □□……□□ / □□……□□□□□



Working Software

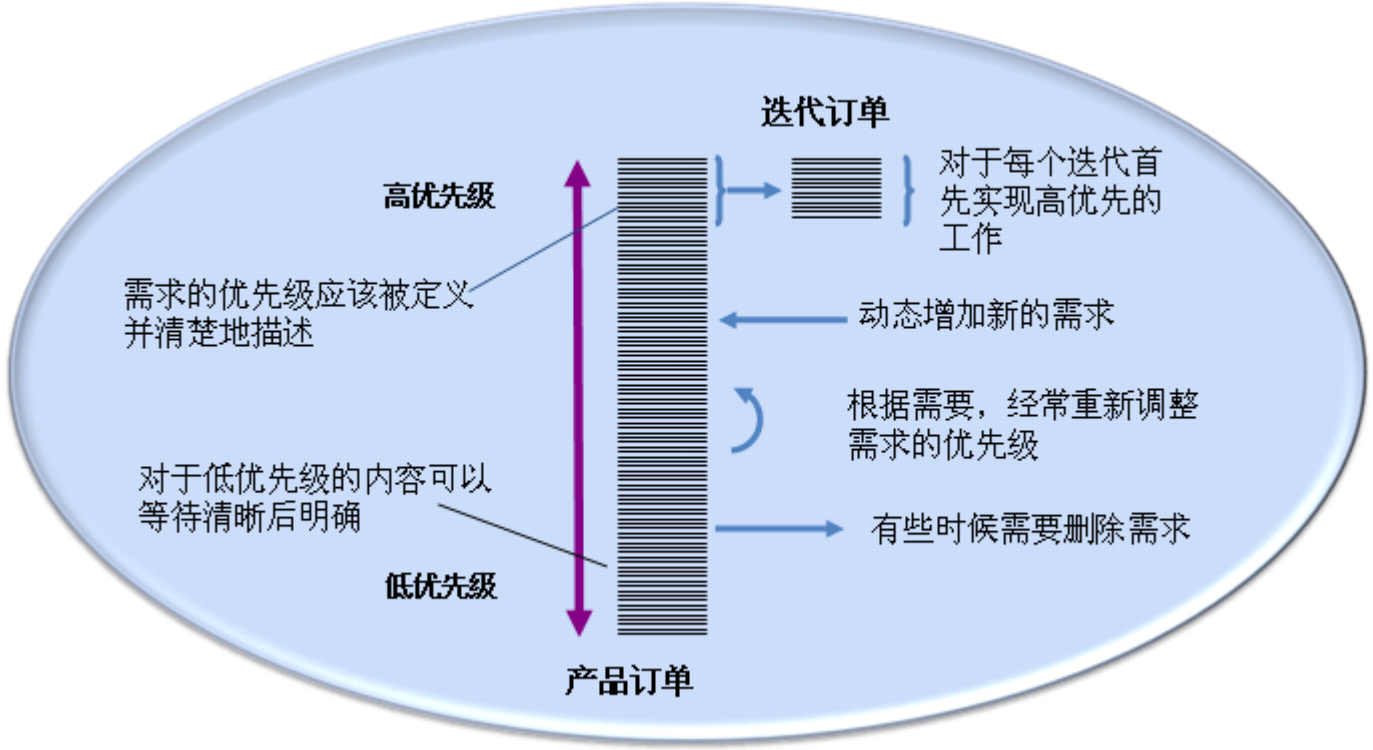
可工作软件是可交付的软件产品。

• “ 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

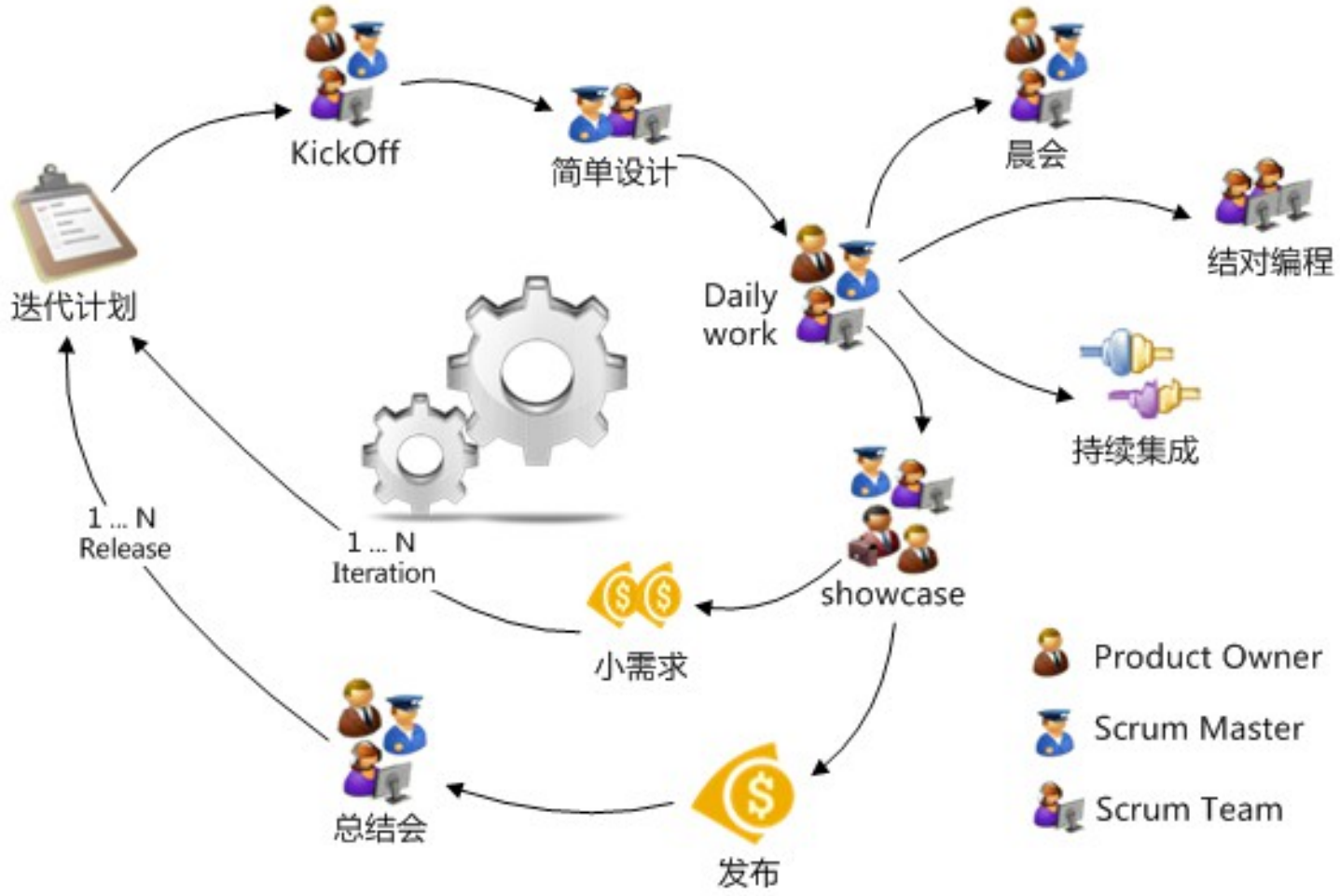
在迭代规划时，产品负责人告诉开发团队需要完成产品订单中的哪些订单项，开发团队决定在下一次迭代中他们能够承诺完成多少订单项。在迭代的过程中，没有人能够变更迭代订单，这意味着在一次迭代中需求是被冻结的。



产品订单

□□□	□□	□□□□□□
1	□□□□	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□①□□□□□②□□□□□□□□□
2	□□□□	□□
3	□□□□	□□
4	□□□□□	□□
5	□□	□□□□□□□□□□□□□□□□□□□□□□□□□□□□
6	□□□□ □□	□□
7	□□□□	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Scrum □□


KickOff
简单设计
晨会
迭代计划
Daily
work
结对编程
持续集成
1 ... N
Release
1 ... N
Iteration
showcase
小需求
总结会
发布
Product Owner
Scrum Master
Scrum Team

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

[illegible]

- [illegible]
- [illegible] 1~2 [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible] Scrum [illegible]

迭代规划会议—第一部分

□□□□□

①□□□□□□□□ Product Backlog □□□□□□□□□□□□□□□□□□□□□□□□□□□□

②□□□□□□□□□□□□

③□□□□□□□□□□□□□□□□□□… …□□

④□□□□□□□

⑤□□□□□“□□”□□□□□□

⑥□□□□□□□□□□□□□□□□□

⑦□□□□□□□□□□□□□□□□□□□□□ UML □□□□□□□□□□□□□□□□

⑧□□□□① □□□□□□□□□□□□□

迭代规划会议—第一部分

□□□□□

- □□□□ Product Backlog □□
- □□ Backlog □□□□□
- □□ Backlog □□□□□□□□□
- □□□□□□□□□□□□□□□□□□□

□□□□□

- □ Sprint □□□
- □□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□□□□□□□□□□□□□□”□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□

迭代规划会议—第二部分

[illegible]

①[illegible] Backlog [illegible]

②[illegible] Backlog [illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- ......

- [illegible]
- [illegible] 10 [illegible]

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

□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

- □□□□□□□□□□□
- □□□□□□ Scrum □□
- □□ 15 □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□“□□□□□□□□□□□”
- □□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□
- Scrum □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- Scrum □□□□□□□□□□□□□□□□□□ Scrum □□□□□□□□□□

每日站立会议

□□□□□

①□□□□□□□□□□□□□□□□

②□□□□□□□□□□□□□□□□□□□□□□□□□□□□

③□□□□□□□□□□□□□□□□□□□

④□□□□□□□□□□□□□□□□□□□□□□□“□□□□□”□□

⑤□□□□□□□□□□□□□□□□□□□□□ Scrum □□□

⑥□□□□□□□□□②□□□⑤□

□□□□□□□□□□□□□□□□

- □□□□□□□□□□□
- □□□□□□□□□□□□□
- □□□□□□□□□□□□

每日站立会议

□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□
- □□□“□□ Backlog”
- □□□“ Sprint Backlog”

迭代评审会议

□□□□□

•Scrum □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Backlog □□□

□□□□□

•□□□□□□□□□□□□□□□

•□□□□□□□□□□□□□□□□□□□□□□□□

•□□□□□□□□□□□□□□□□

□□□□□

•□□□□□□□□□□□□□□

•□□□□□□□□□□□□□□□

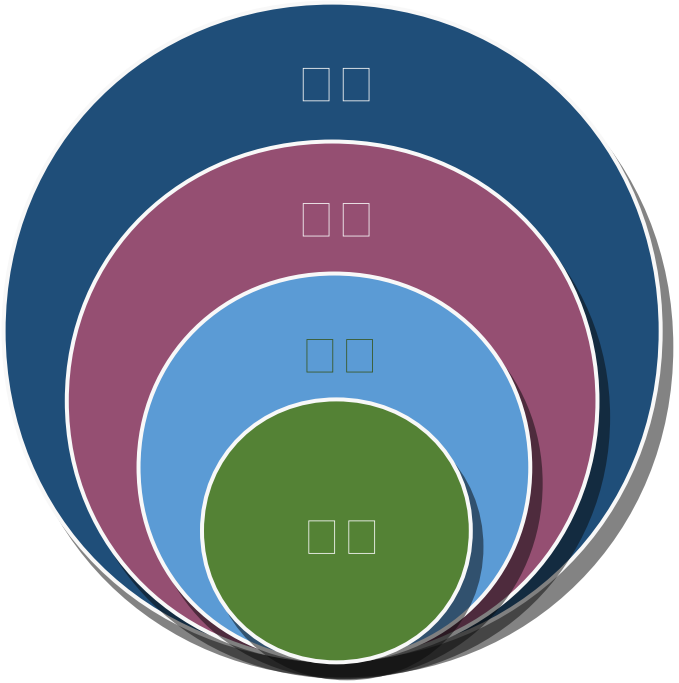
迭代总结会议

□□□
□□□□□□□□□□□□□□□□4□□□

□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

Scrum □□



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。

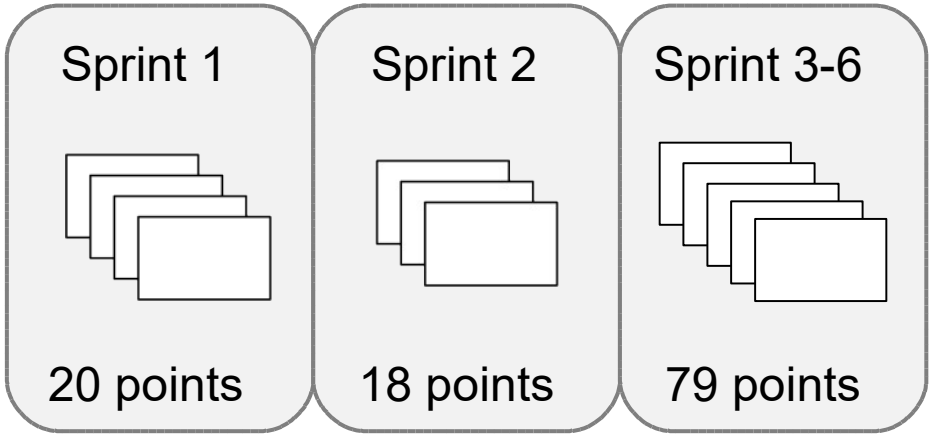
Scrum □□

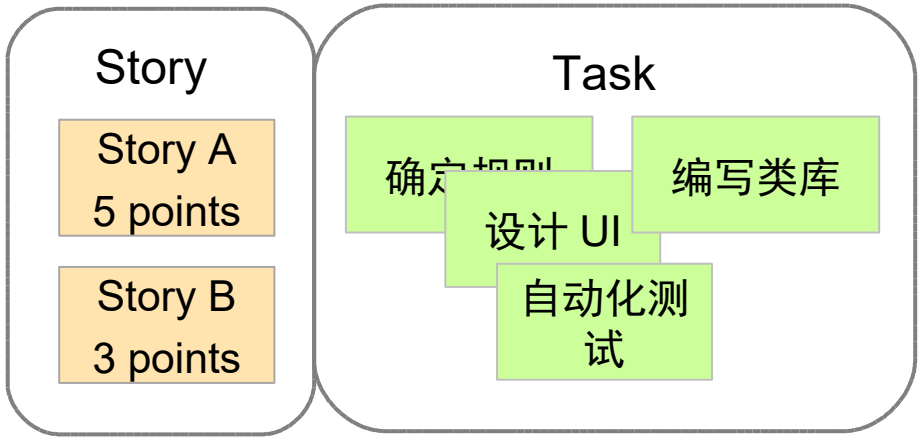
□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□

□□□□

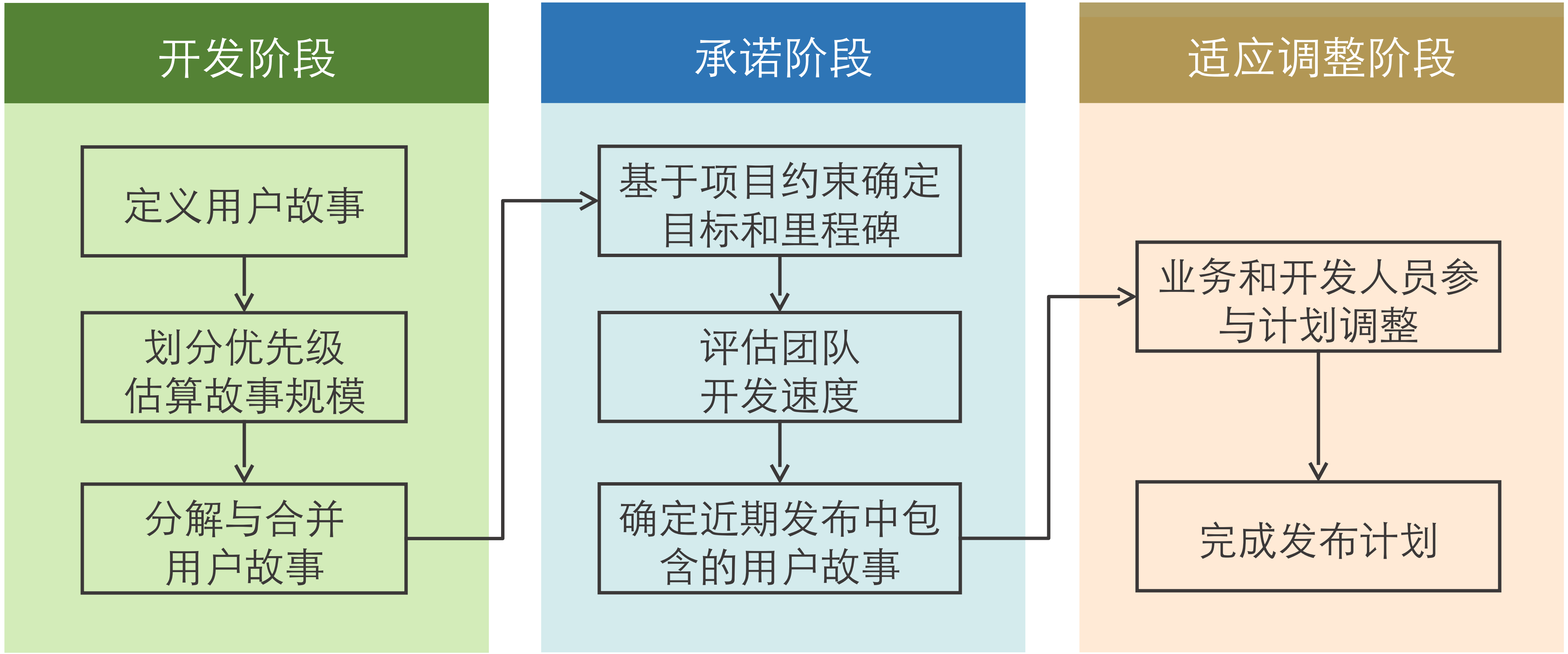
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□


Sprint 1
20 points
Sprint 2
18 points
Sprint 3-6
79 points


Story
Story A
5 points
Story B
3 points
Task
确定规则
编写类库
设计 UI
自动化测试

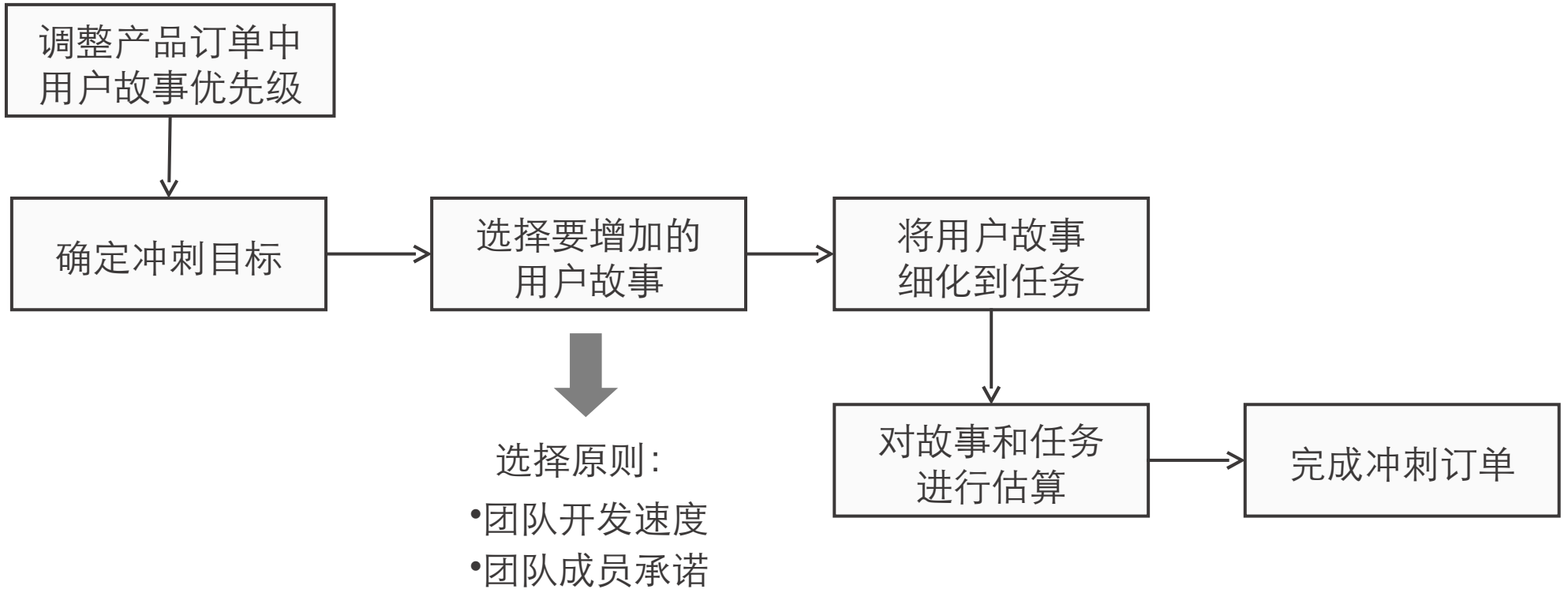
Scrum □□

□□□□□□□□□□□□□□□□□□□□□□□□□□ 3 □ 6 □□□


开发阶段
定义用户故事
划分优先级
估算故事规模
分解与合并
用户故事
承诺阶段
基于项目约束确定
目标和里程碑
评估团队
开发速度
确定近期发布中包
含的用户故事
适应调整阶段
业务和开发人员参
与计划调整
完成发布计划

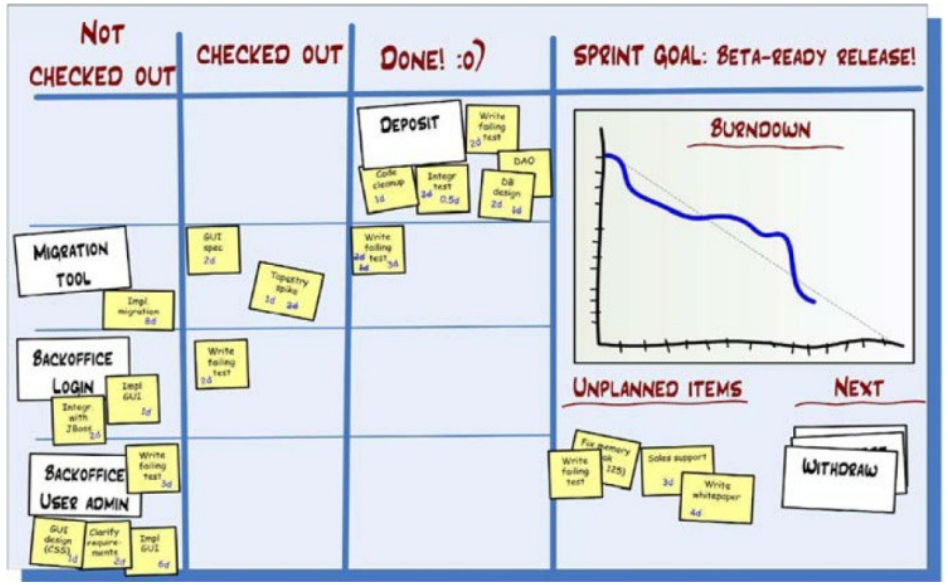
Scrum □□

□□□□□□□ Scrum □□□□□□□□□□□□□□□□ 4 □□□□□



可视化管理

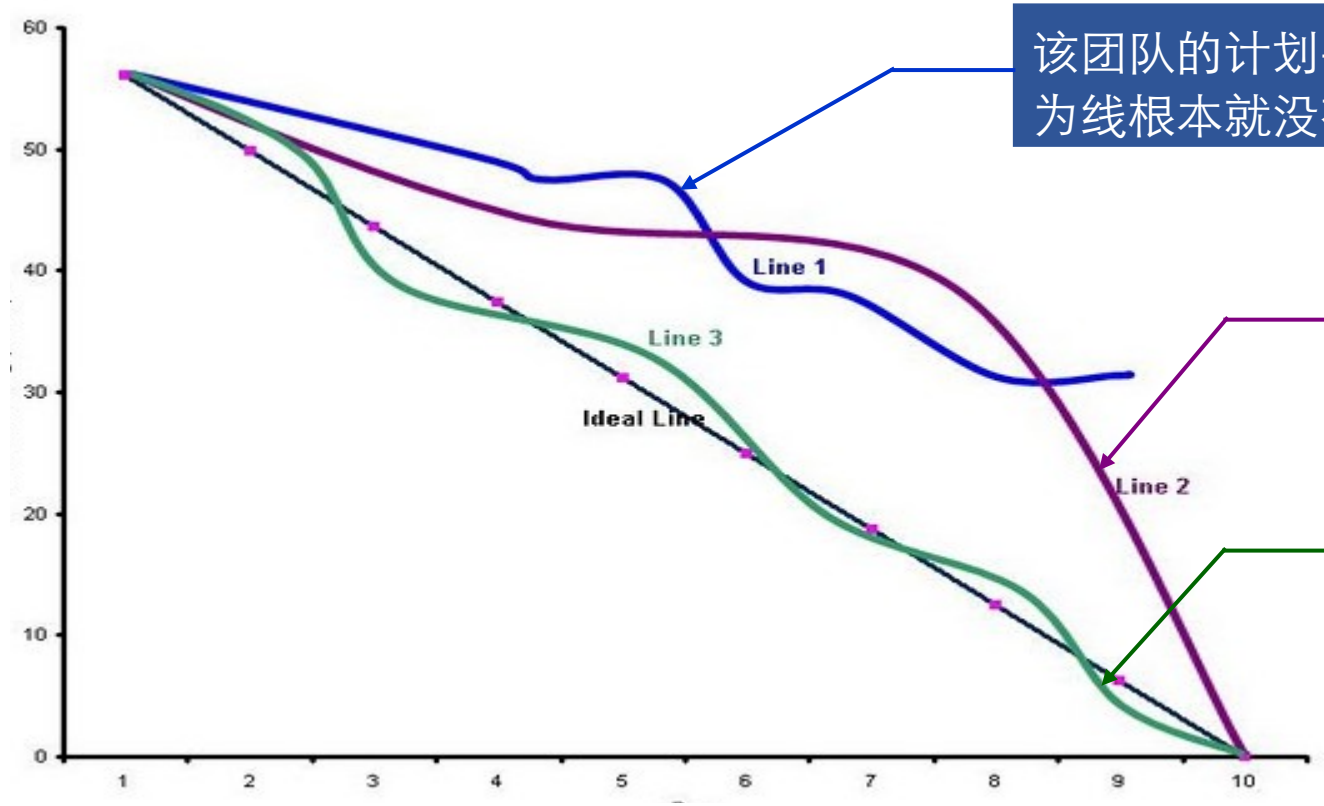
任务白板是团队开发的晴雨表，它将团队的任务和进度可视化地展现出来。而引入电子白板可能会削减团队之间的沟通，降低团队的透明度，违背了敏捷重视人和团队的原则。



可视化管理

□□□□□□□□□□□□□□□□□□□□□ Y □□□□□□ X □□□□□□

□
□
□
□
□



该团队的计划并不好，因为线根本就没有触到零点

该团队已经达成了目标，但并没有主动去更新数字

该团队计划良好，实现了足够故事，接近理想情况

□□

教学提纲

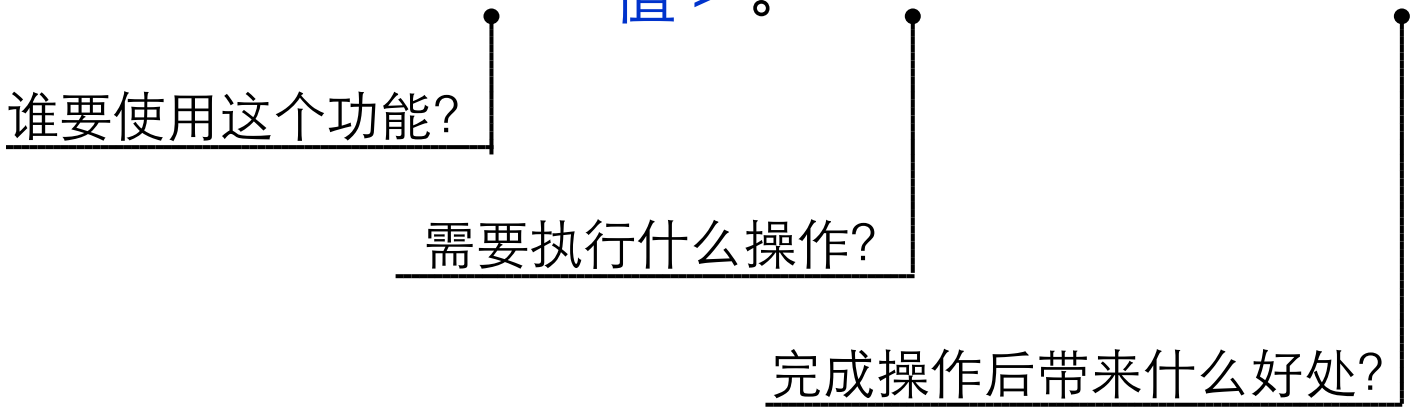
3 □□□□□□

- □□□□□□□□
- □□□□
- □□□□□□
- □□□□

用户故事

□□□□□ User Story □□□□□□□□□□□□□□□□□

格式：作为一个 < 角色 > ，可以 < 活动 > ，以便于 < 价值 > 。



用户故事

[illegible]	[illegible]	[illegible]
1	[illegible]	[illegible]①[illegible]②[illegible]
2	[illegible]	[illegible]
3	[illegible]	[illegible]
4	[illegible]	[illegible]
5	[illegible]	[illegible]
6	[illegible]	[illegible]
7	[illegible]	[illegible]

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。

- □□
- □□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 60-80% □□□□□□□□□□□□□

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	[illegible]① [illegible]② [illegible]	10
2	[illegible]	[illegible]	15
3	[illegible]	[illegible]	1
4	[illegible]	[illegible]	40
5	[illegible]	[illegible]	28
6	[illegible]	[illegible]	12
7	[illegible]	[illegible]	7

敏捷估算

[illegible] Delphi [illegible]

- [illegible]
- [illegible] 10 [illegible] 10 [illegible]
- [illegible]

敏捷估算

[illegible] Joker [illegible]

[illegible] 54 [illegible] 4 [illegible] 13 [illegible] Joker [illegible]

[illegible] 13 [illegible] 1/2 [illegible] 1~10 [illegible] 20 [illegible] “[illegible]” [illegible]

- [illegible] 4 [illegible]
- [illegible] 4-8 [illegible]

敏捷估算

□□□□□□□□□□□□□ Backlog □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□

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。

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。

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后，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。

[illegible] Scrum Master [illegible]

简单设计

□□
□□□□□□□□□□□□□□□□□□□□□

- 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

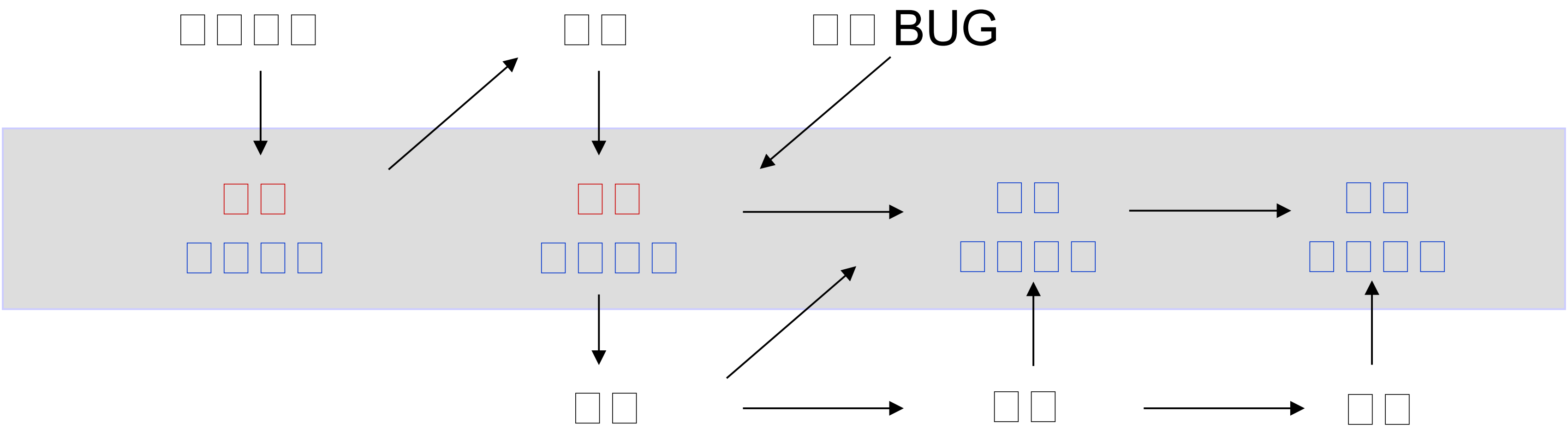
对简单设计的需求并不是说所有设计都很小，也不表示它们是无足轻重的；它们只不过需要尽可能地简单，但是仍能工作。

结对编程

[illegible]

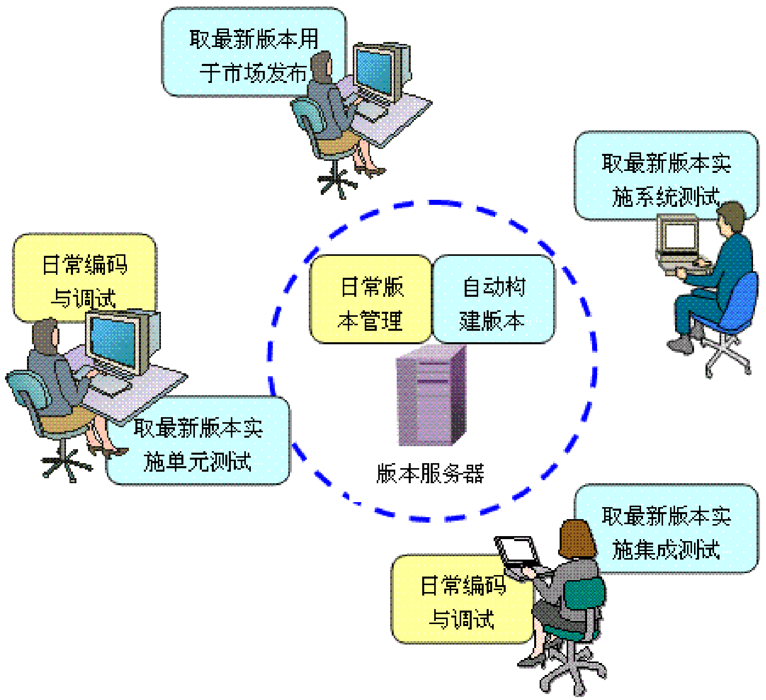
- [illegible]
- [illegible] 15% [illegible]
  15% [illegible]
- [illegible]
- [illegible]

测试驱动开发



持续集成

□□
□□□□□□□



- 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。

教学提纲

4 □□□□□□□□

- □□□□□□□□
- □□□□□□□□
- Popush □□□□
- □□□□□□□

谢谢大家！

THANKS