



思维导图

敏捷意义和敏捷生命周期

传统项目管理VS敏捷项目管理

斯泰西图

根据需求和技术的确定程度，划分区域，选择合适的管理方式。

敏捷三角形

价值、质量、约束

四种生命周期类型

预测型，需求确定范围确定，交付频率低

迭代型，需求动态，范围不确定，交付频率低

增量型，需求迭代固定，整体动态，交付频率高

敏捷型，需求动态，范围不确定，交付频率高

混合型生命周期

既有预测又有敏捷，或者处于过渡阶段，都可以采取混合生命周期。

根据变更的程度和交付的频率，判断预测和敏捷谁为主谁为辅。



思维导图



敏捷宣言

个体以及互动 胜过
流程和工具

以人为本

流程是死的，人是活的

可用的软件 胜过
完整的文档

当相关方想要详尽的文档、详细的计划时，可以展示成功经验说服。

可用的软件胜于完整的文档，但不是完全不需要文档。

当详细的文档已经是需求的一部分，我们需要关注价值，做出调整。

客户合作 胜过
合同谈判

不一味遵循合同，与客户意见相左时，强调沟通合作达成一致，追求合作双赢。

应对变更 胜过
遵循计划

计划、需求等都可能发生变化，应该积极应对。

拥抱变化不等于无条件接受所有变化。



思维导图



敏捷十二大原则

工作原则

9.对技术卓越和好的设计的持续关注有助于增强敏捷性。

10.尽量做到简洁，尽最大可能减少不必要的工作。这是一门艺术。

12.团队要定期回顾和反省如何能够做到更有效，并相应地调整团队的行为。

团队原则

11.最佳的架构、需求和设计出自自组织团队。

5.要善于激励项目人员，给他们所需要的环境和支持，并相信他们能够完成任务。

4.在项目过程中，业务人员与开发人员要始终通力合作。

6.团队内部和各个团队之间，最有效的沟通方法是面对面的沟通。

1.我们的最高目标是，通过尽早持续地交付有价值的软件来满足客户的需求。

3.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。

8.敏捷过程提倡可持续的开发。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。

7.可工作软件是衡量进度的首要指标。

2.即使在项目开发的后期，仍欢迎对需求提出变更。敏捷过程通过拥抱变化，帮助客户创造竞争优势。



构想阶段实践

(希赛)

敏捷项目管理 阶段框架-构想阶段

商业论证

愿景

通过产品愿景盒、电梯测试获取，记录在项目章程中。颗粒度是大的、粗略的、高层级的。

项目章程

粗略的、大颗粒的

团队章程

团队需共同遵守的行为准则



敏捷项目管理阶段框架

构想

- 商业论证
- 愿景^①
- 项目章程^①
- 团队章程^①

推测

- 时间盒^①
- 洋葱圈规划（滚动式规划）^{①⑦}
- 用户故事^{③③}
- 风险调整待办事项
- 刺探/探针^②

探索

- 迭代计划和监督
- 技术实践^⑥
- 教练和团队发展^⑤

适应

- 客户评审
- 团队绩效评估
- 速度^⑮
- 项目整体状态
- 根本原因分析

结束

- 回顾



小结-洋葱圈



洋葱圈规划 (滚动式规划)

愿景

产品的简要概述，未来状态、高层级大颗粒的，记录在项目章程中

产品路线图

产品可视化概述，大概的时间阶段和功能描述

用户旅程地图

用户故事地图

依据客户价值，由用户故事构建的故事地图

MVP

能运行满足客户最主要需求的最小架构

MMF

在运行的基础上添加的最小单位的完整功能包

发布计划

包含大颗粒度版本和大致时间信息，一个版本发布意味着推出市场上市

迭代计划

一个迭代需要完成的内容

每日计划

一天需要完成的任务



小结-用户故事



性能需求卡片

用户故事

起源

用户故事这个词起源于极限编程XP

构建场景和人物

主要是为了收集需求，描述用户群体（目标群体）需求

3C原则

卡片、交谈、确认

INVEST原则

独立的、可协商的、有价值的、可估计的、小的、可测试的

DOD

验收的标准

在用户故事被确定好的时候明确

故事点概念

比较用户故事的规模、复杂程度、不确定性，不容易跨团队比较

优先级排序方法

必须做的、应该做的、可以做的、不要做的

MOSCOW法则

基于用户需求，用户满意进行分类和优先排序

卡诺分析

依据价值和风险进行象限区分，优先处理高价值高风险的内容

风险四象限

100点法

举手表决法

优先级排序依据

价值最重要，主题风险成本政治等也会有影响

故事点估算

卡牌、公开讨论、异常值着重讨论、多轮达成共识

计划扑克

匿名、专家、多轮、提交个人估算

宽带德尔菲

不考虑速度和中断

理想时间

团队成员一起进行估算，团队成员意见最重要



推测阶段实践

敏捷项目管理 阶段框架-推测阶段

洋葱圈规划（滚动式规划）

17

用户故事

33

风险调整待办事项

刺探/探针

刺探是对风险或潜在问题进行探索性工作，探索新技术，新方法，新问题，不清楚的风险

当客户需求不断变化时，可以用刺探迭代来分析客户需求

时间盒

固定的一段相对比较短的时间，计划的工作要在这段时间内完成



探索阶段



敏捷项目管理 阶段框架-探索阶段

迭代计划和监督

技术实践

低成本变更

技术债务

简单设计

持续集成

自动化测试

重构

教练和团队发展

教育客户

构建信任

互动

冲突解决

参与式决策



适应阶段实践



敏捷项目管理 阶段框架-适应阶段

客户评审

团队绩效评估

速度

- 概念
本次迭代中实际完成功能的故事点大小的总和
- 计算
已完成的故事点才算作速度
- 初始速度估计
初始速度估计参考过去，或以往的经验，以往的平均速度
- 特征
追求速度稳定、连续一致，最好的结果是速度达到持续稳定，不能跨团队比较
- 展现
燃起图燃尽图
 - 颗粒度考察
 - 图表变化
 - 风险燃尽图
- 监控
与预期不符时，以团队实际速度为主，通过减少或增加故事点数以适应团队实际速度

项目整体状态

根本原因分析



Scrum敏捷实践



从相关方处获得需求



产品负责人 (Product Owner)

1
2
3
4
5
6

产品待办列表
(Product Backlog)



团队

团队选择该迭代
承诺完成多少工作

迭代计划会议

下个迭代版本
产品待办列表整理



迭代待办事项
(Sprint Backlog)



敏捷教练
(Scrum Master)



每日站会



迭代评审会议



潜在可发布的产品增量



迭代回顾会议

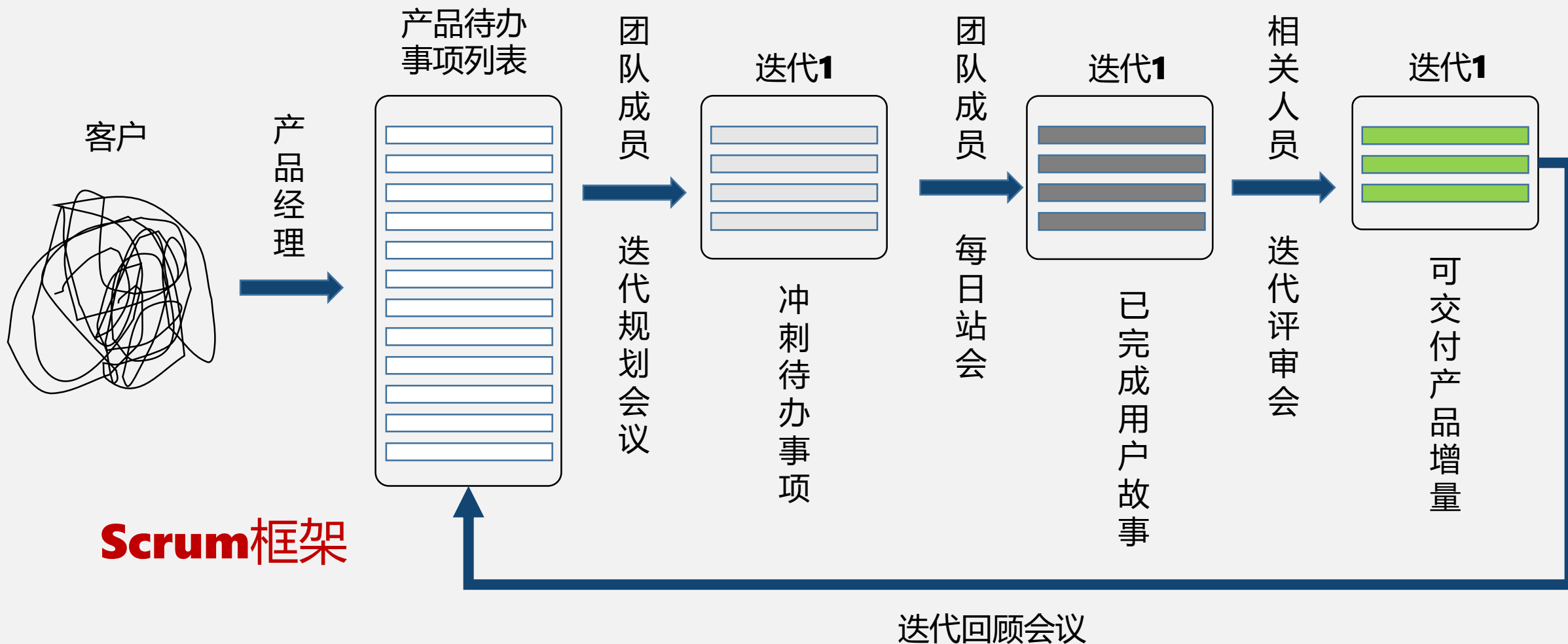


迭代
(1-4周)

在一个迭代版本中
目标没有发生变化



Scrum敏捷实践





Scrum敏捷实践

三大支柱

透明、检查、适应

三个角色

产品负责人

敏捷教练

自组织团队

三个工件

产品待办事项列表

迭代待办事项列表

可交付产品增量

五个事件

迭代、冲刺

冲刺计划会议

每日站会

迭代评审会议

迭代回顾会议

五大价值观

承诺、专注、开放、尊重、勇气



迭代规划会议

做什么

冲刺计划会议是确定迭代任务和目
标，讨论如何完成，产出迭代待办事
项列表

谁参加

PO、开发团队、敏捷教练都需要参会
当相关方想了解团队计划，了解需求
目标变化时，可以受邀参与冲刺计划
会议

开多久

时间盒2X，一个月的迭代周期冲刺计
划会议时间盒为8小时



迭代回顾会议

做什么

回顾过程、找出原因、提出改进并排序

开多久

迭代评审之后，下次迭代规划之前

一个月的迭代一般是3小时

谁参加

一般是团队成员、敏捷教练，PO视情况，相关方受邀参与



迭代评审会议

做什么

迭代评审会议上PO鉴定“已完成”，客户验收已完成的用户故事，给出反馈

讨论下次预期和趋势

未完成或者未验收的用户故事需要协商沟通后返回产品待办事项列表

开多久

一个月的迭代周期会议时间盒是4小时

谁参加

产品负责人、主要客户、敏捷教练、开发团队



每日站会

做什么

每日站会了解项目状态信息（掌握进度情况），了解团队个人状态，共识信息，发现问题、障碍、风险

每日站会是支撑敏捷原则最重要的基石之一，不能随意取消

怎么开

每日站会是团队自行组织，时间地点都是自行讨论确定，任何人都可以主持站会

开发团队和敏捷教练都参会，PO视情况，相关方受邀。

时间盒一般15分钟

因为其他原因（文化环境等）影响到会议效果，可以适当延长时间，但是一般不超过20分钟

人数过多，可以考虑划分团队

反模式

不是状态报告会议

只发现问题，不解决问题

问题可以站会后专门讨论解决

如果是不影响当前迭代目标，和整体流程相关的，可以回顾会议讨论改进



DEEP模型，详略适宜的、可估计的、涌现的、排好优先级的

产品待办事项列表的本质是所有工作的有序列表，以价值为导向

主要由PO创建并维护

产品待办事项列表的排序考虑价值+风险，还可能考虑成本、依赖、政治等

产品待办事项列表

包含功能性内容和非功能性内容，技术债务、风险应对、运维工作等作为非功能性用户故事列入待办事项

产品待办事项列表的细化梳理是渐进明细的



迭代待办事项列表

迭代待办事项列表本质是本次迭代需要完成的工作项

在迭代规划会议上创建

迭代过程中迭代待办事项列表一般不会随意改变，我们首先要确保迭代任务的顺利完成

如果用户故事已经显然无效，或者当前情况紧急（不处理会危急整个迭代），可以对迭代待办事项列表进行调整



产品负责人

对接客户（发起人），收集需求，搞定客户（发起人）

PO职责—聚焦于产品

创建（或与团队共同创建）产品待办事项列表

根据商业价值对任务进行排序（高于团队成员）

监控需求，根据实际情况，清理、变更需求及排序

参与项目，经常及时给出反馈，鉴定“已完成的用户故事”



敏捷教练

为他人的贡献铺路，提供支持和帮助，协助团队解决问题，确保敏捷实践的效果

敏捷教练清除组织障碍，不干涉产品开发方向和如何开发的问题

仆人式领导，角色定位是辅助性、仆人式、服务型

敏捷教练促进作用，促进团队内外合作交流，不会替他人做决定

敏捷教练插手：被求助时、团队违反敏捷时、不会用敏捷工具、冲突自行解决无效



团队成员更换，一般团队形成后不会随意更换增减人员，如果不得不进行更换，比较好的时机是两个迭代之间进行

团队建设的五阶段，形成、震荡、规范、成熟、解散

首选集中办公，促进渗透式沟通，分布式团队可以用虚拟手段，鱼缸窗口和远程结对。

自组织团队

聚焦绩效，自我管理，自行认领任务，建设性对抗，不对敏捷教练承诺

3-9人，正负2变动，人数过多可以拆分团队。

专职，专注于一个项目

通才型专家，有利于减少瓶颈、提升效率、减少孤岛/竖井



识别一个工作项中各个流程的流动过程，找出其中的缺陷（浪费）并制定改进措施，（找到不能为产品提供增值的活动）





看板

作用是可视化工作流程，共识信息和风险

限制在制品

合理规划、团队高效、找出瓶颈

累积流图

基于看板不同模块的完成情况统计，
判断当前项目的范围变化和趋势



极限编程

开发的极限、反馈回路的极限

结对编程

两个开发人员，一起工作，一个写代码一个实时审查

作用是更专注高效、破除团队壁垒、培养新技能

半私密空间（洞穴和公共区域）

持续集成

立刻集成，便于迅速发现代码问题并解决问题

测试驱动开发

在开发之前写好测试用例，一边编代码一边测试，测试不通过的可以重构

自动化测试

避免交付后故障频繁，避免人工测试增加成本

重构

在不改变功能的情况下对代码进行优化重组，提升代码的可维护性