



敏捷项目管理



第一章 敏捷价值及敏捷生命周期 <3%

第二章 敏捷宣言及十二原则 5%

第三章 敏捷项目管理阶段框架 10%

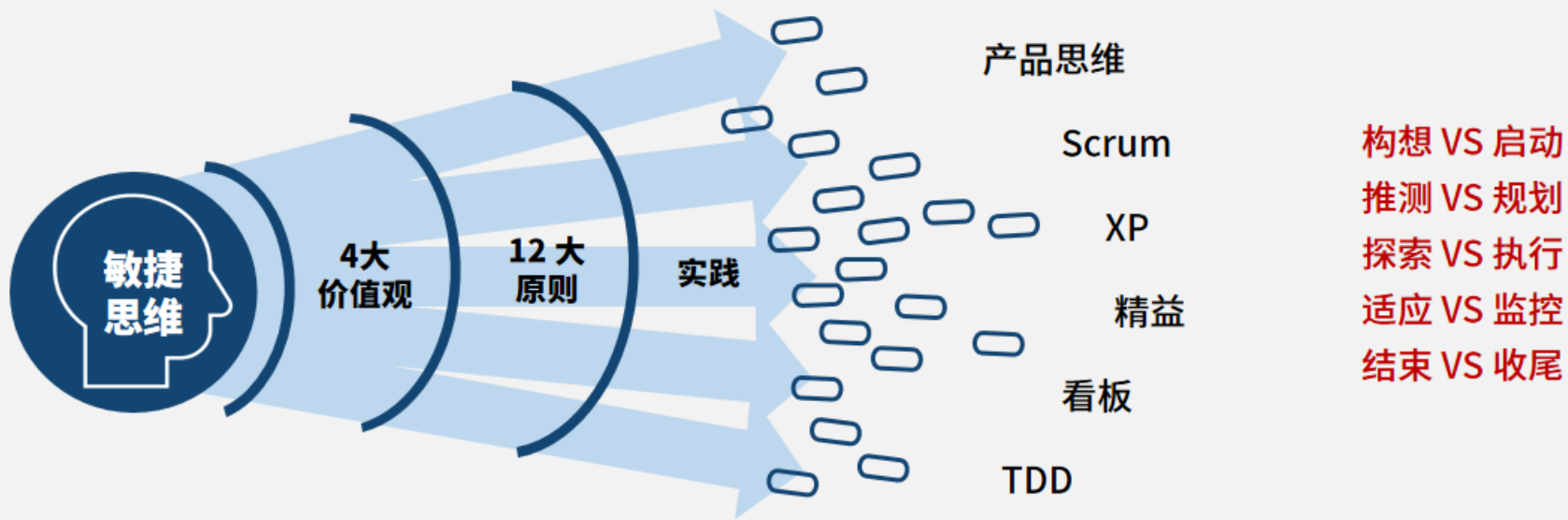
第四章 Scrum实践 28%

第五章 其他实践（精益、看板、XP、TDD） 5%

第六章 生命周期类型选择及变革模型 <2%



敏捷项目管理



敏捷思维模式由价值观定义，以原则为指导，并在许多不同的实践中体现。

敏捷实践者根据自身需求选择不同的实践。



敏捷宣言



考点：敏捷宣言★

我们正在通过亲自开发和帮助他人开发，发现开发软件的更好方法。通过这项工作，我们开始更重视：

个体以及互动 而不是 流程和工具

以人为本

可用的软件 而不是 完整的文档

以结果为导向

客户合作 而不是 合同谈判

合作共赢

应对变更 而不是 遵循计划

拥抱变化

也就是说，右栏中的项目固然有价值，但我们更重视左栏中的项目。

2001年2月，17位来自软件开发领域，敏捷各流派的领军人物共同制定



敏捷宣言一十二原则

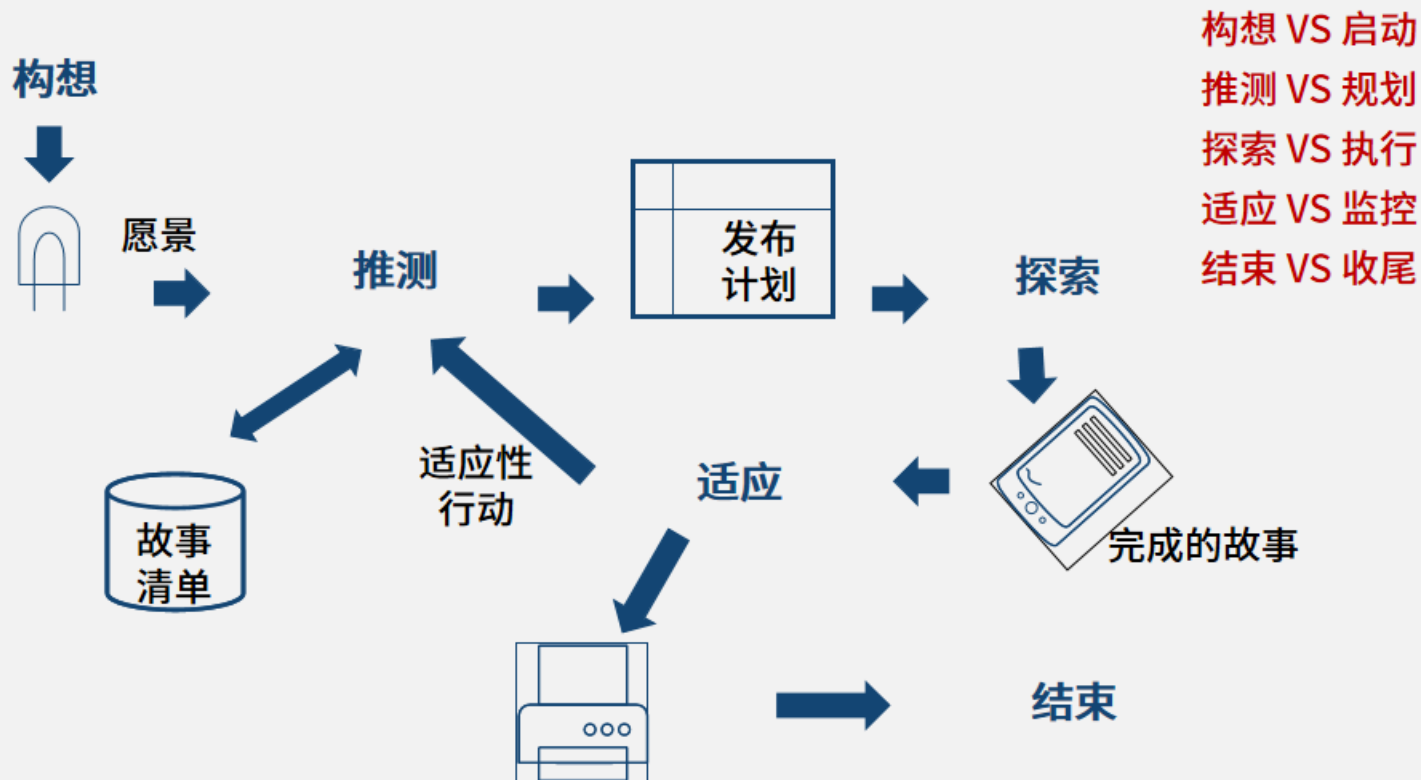
价值驱动交付



- (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**。
- (2)欢迎对需求提出变更，即使在项目开发的后期也不例外。敏捷过程要善于利用**需求变更**，帮助客户获得竞争优势。
- (3)要**经常交付**可用的软件，周期从几周到几个月不等，且越短越好。
- (4)项目实施过程中，业务人员与开发人员要必须始终**通力协作**。
- (5)要善于激励项目人员，给予他们所需的**环境和支持**，并相信他们能够完成任务。
- (6)无论是对开发团队还是团队内部，信息传达最有效的沟通方法都是**面对面的交谈**。
- (7)**可工作软件**是**衡量**进度的首要**指标**。
- (8)敏捷过程提倡**可持续的开发**。项目发起人、开发人员和用户应该都能够始终保持**步调稳定**。
- (9)对**技术的精益求精**以及对设计的不断完善将提高敏捷性。
- (10) **简洁**，即尽最大可能减少不必要的工作，这是一门艺术。
- (11)最佳的架构、需求和设计将出自于**自组织团队**。
- (12)团队要**定期反省**怎样做才能更有效，并相应地调整团队的行为。



敏捷项目管理阶段框架





敏捷项目管理阶段框架



五个阶段的实践清单

构想阶段

项目论证
项目章程
产品愿景说明
电梯测试说明
产品架构
项目数据表
得到正确的人员
参与人员识别
客户和团队接口
过程定制

推测阶段

场景和人物角色
时间盒、功能
洋葱圈、产品路线图
版本、里程碑、迭代计划
用户故事、DOD、3C原则、
INVEST原则、故事点估算
MoSCoW、卡诺分析、
风险四象限、刺探、
性能需求卡片
产品待办事项

探索阶段

工作量管理
低成本变更
教练和团队发展
每日团队会议
持续集成
重构
冲突解决
参与式决策
客户每日交互

适应阶段

迭代审查
迭代回顾
速度
燃尽图
累积流图

结束阶段

项目总结
复盘
庆祝



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)



迭代
(1-4周)

在一个迭代版本中
目标没有发生变化



每日站会



迭代评审会议



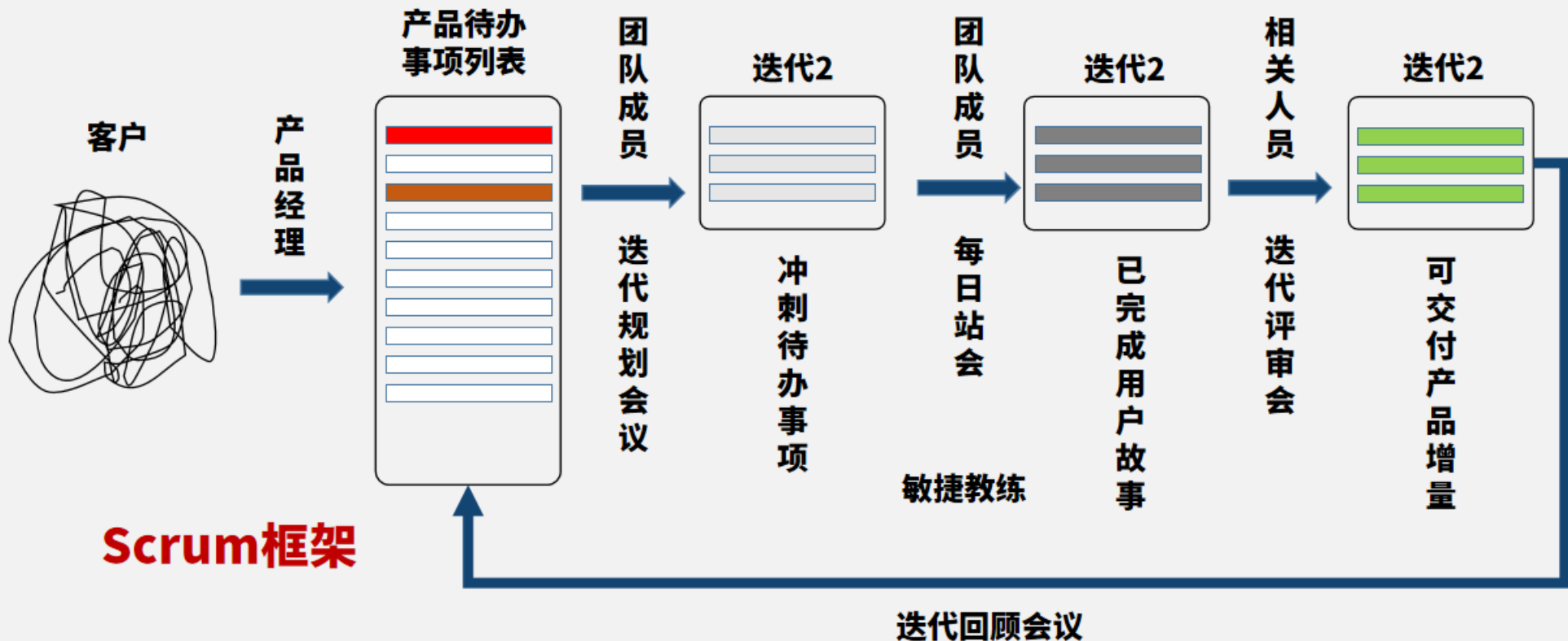
潜在可发布的产品增量



迭代回顾会议



Scrum敏捷实践





Scrum敏捷实践



考点：Scrum框架中的33355

三个支柱	三个角色	三个工件
透明性 (Transparency)	产品负责人 (Product Owner)	产品待办事项列表 (Product Backlog)
检视 (Inspection)	敏捷教练 (Scrum Master)	冲刺待办事项列表 (Sprint Backlog)
适应 (Adaptation)	项目团队 (Scrum Team)	可交付产品增量 (Increment)



Scrum敏捷实践



考点：Scrum框架中的33355

五个事件	五大价值观
冲刺/迭代 (Sprint)	承诺 (Commitment) - 愿意对目标做出承诺
冲刺规划会议 (Sprint Planning)	专注 (Focus) - 全身心都用到你承诺的工作上去
每日站会 (Daily Scrum)	开放 (Openness) - 团队内所有信息对所有人开放
迭代评审会议 (Sprint Review)	尊重 (Respect) - 每个人都有他独特的价值和经验
迭代回顾会议 (Sprint Retrospective)	勇气 (Courage) - 勇于承诺，履行承诺，敢于说不



精益思想



考点：精益七大核心理念★

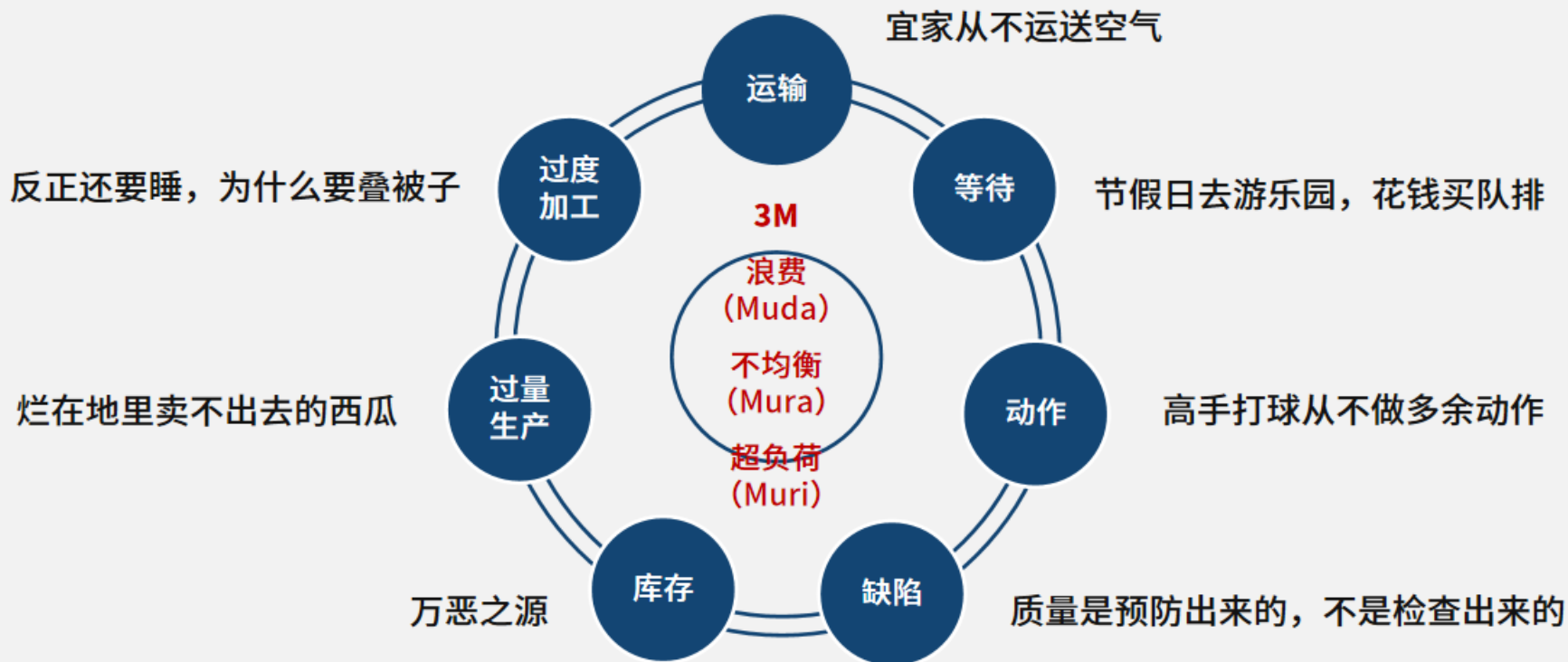




精益思想

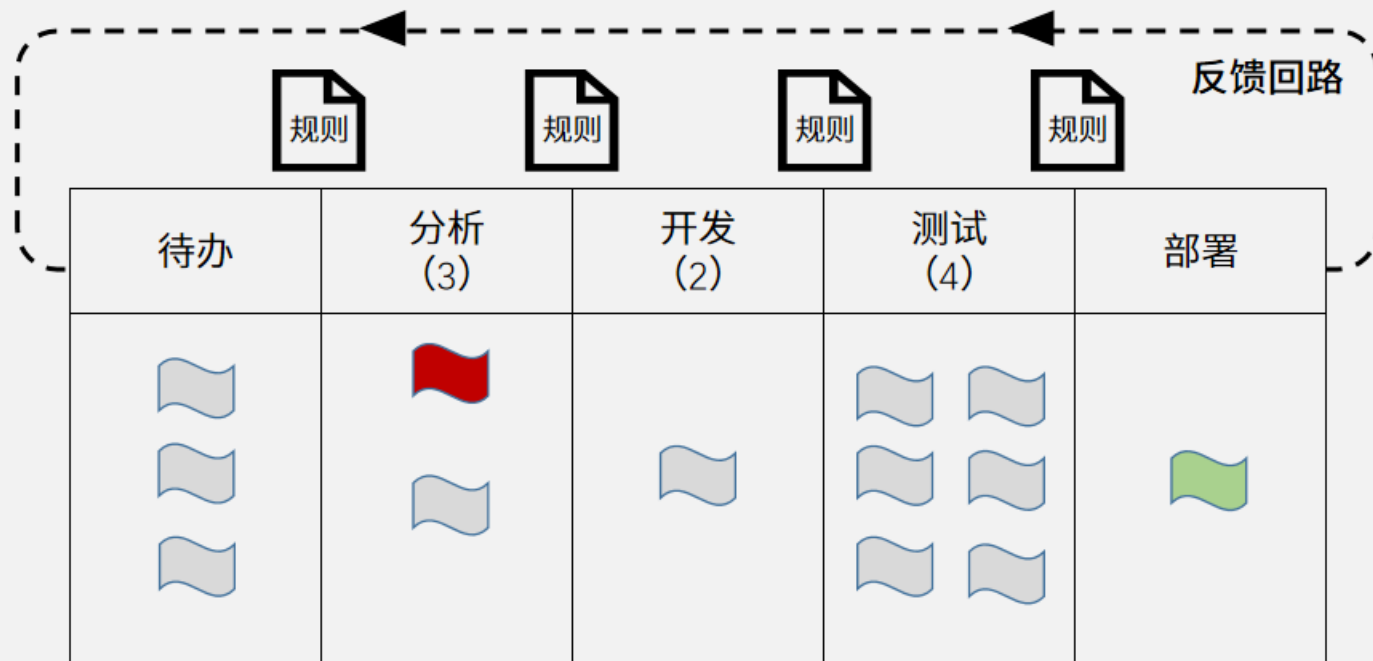


考点：精益制造七浪费★





看板实践 (Kanban)



任务板 (kanban)

看板六大核心实践

- 1、可视化工作流程
- 2、约束在制品 (WIP)
- 3、度量和管理工作流 (拉动)
- 4、显示化规则
- 6、建立反馈环
- 6、在协作及实验中改进



极限编程XP



- 极限编程 EXtreme Programming (XP) 是一种基于频繁交付周期的软件开发方法。
关注团队凝聚力、沟通、代码质量和编程。

XP实践领域	主要	次要
组织	<ul style="list-style-type: none">• 集中办公• 整个团队• 信息灵通的工作场所	<ul style="list-style-type: none">• 真实客户参与• 团队连续性• 可持续节奏
技术	<ul style="list-style-type: none">• 结对编程• 测试驱动开发• 增量设计	<ul style="list-style-type: none">• 共用代码/集体所有制• 代码和测试文档• 重构
规划	<ul style="list-style-type: none">• 用户故事• 每周周期• 每季周期• 松弛	<ul style="list-style-type: none">• 根本原因分析• 裁剪团队• 按使用情况支付• 协商范围合同• 每日站会
整合	<ul style="list-style-type: none">• 10分钟构建• 持续集成• 测试优先	<ul style="list-style-type: none">• 单代码库• 增量部署• 每日部署