希赛

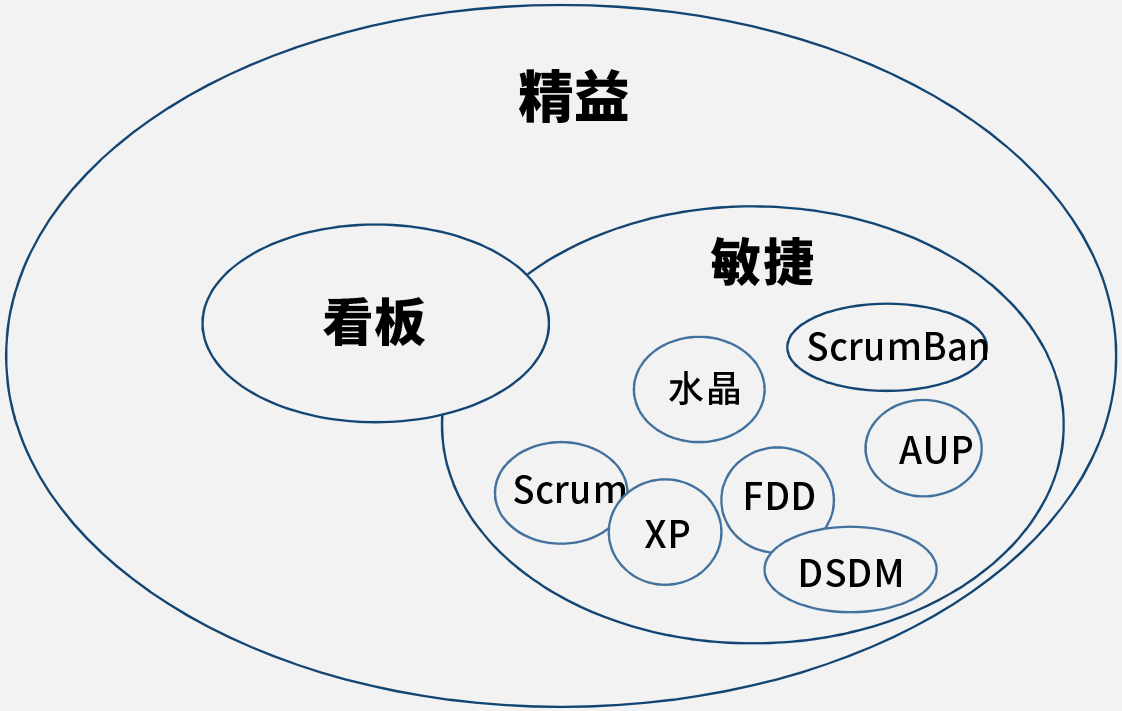
敏捷项目管理

《其他敏捷实践》

敏捷项目管理

希赛

敏捷是许多方法的总称



其他敏捷实践
常见实践
精益Lean
看板Kanban
极限编程XP
其他方法
水晶方法
Scrum Ban
FDD
DSDM方法
AgileUP
扩展框架
Scrum of Scrums
LeSS
SAFe®
企业Scrum
规范敏捷DA

希赛

精益（Lean）

精益思想

希赛

精益思想

精益思想就是花最少的钱，办最好的事，核心是消除浪费。

浪费3M

不均衡（Mura）

超负荷（Muri）

浪费（Muda）

由于异常情况不均衡（Mura）出现，如客户订单突然增加了，导致了工厂的产能超负荷（Muri）了。工人为了赶订单，匆忙之中出错的概率增加，引起次品率上升，浪费（Muda）于是就诞生了。

精益思想

考点：精益七大核心理念★

精益思想

希赛

考点：精益七大核心理念★

消除浪费	要想获得最大化的价值，必须将浪费最小化。延期、没有用的功能、等待都是一种软件浪费。
授权团队	尊重团队成员并由团队来进行决策，保证项目的效率并且有利于项目成功。
快速交付	通过快速交付有价值的软件来最大化项目的投资回报率（ROI）。
全面优化	去检视系统的整体而不是一部分，关注整个组织的优化改进，关注团体。

精益思想

希赛

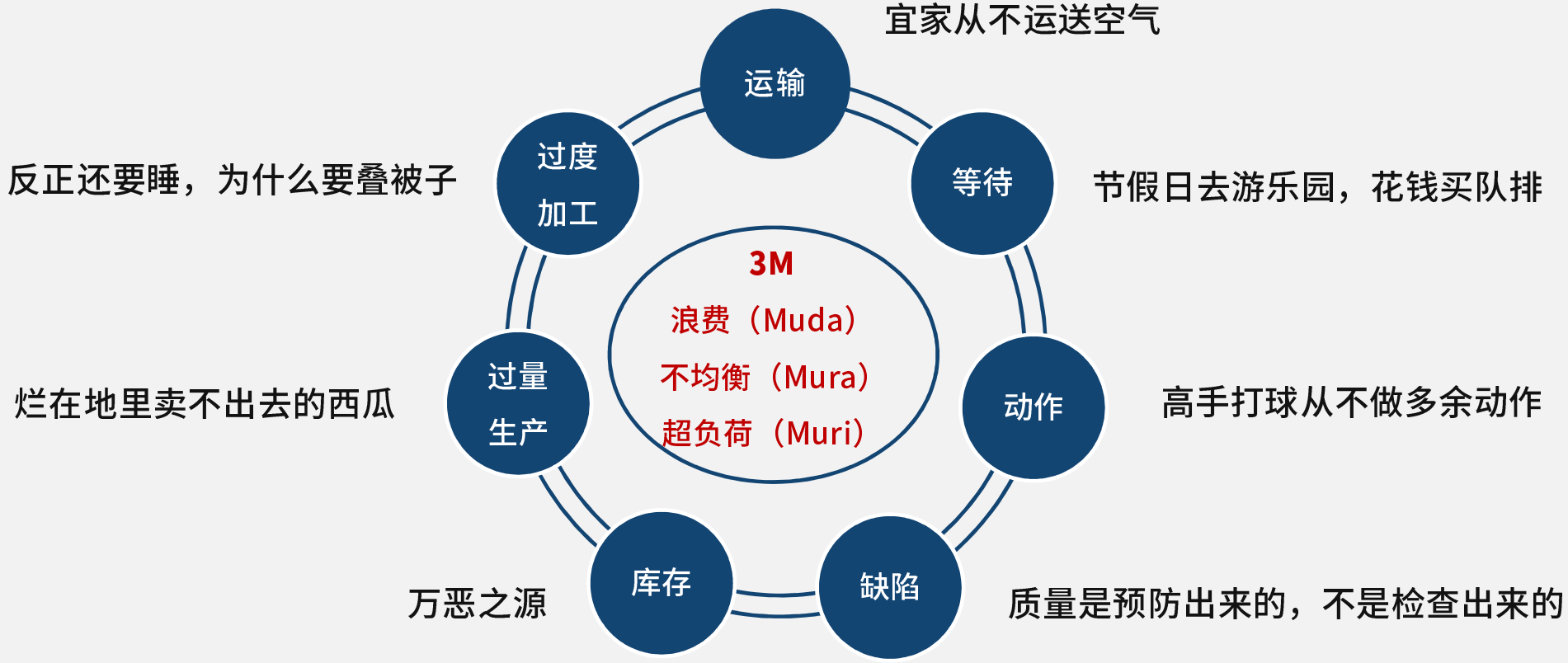
考点：精益七大核心理念★

品质为先	精益开发不是测试为先，而是通过重构、持续集成、单元测试等技术手段来加强质量保证。
晚做决策	尽早计划和最晚决策。
强化学习	通过尽早沟通和频繁的反馈来建立学习的内容。软件项目是业务和技术两项经验的积累，需要尽快开始并保持学习的状态。

精益思想

希赛

考点：精益制造七浪费★



精益思想

希赛

考点：消除浪费★

精益思想关注的三大低价值事件：Muda(浪费)，Mura(不均衡) 和Muri(超负荷)

6吨货物

卡车负载最多4吨

仓库工人3人，同时处理最多3吨货物

精益思想

希赛

考点：消除浪费★

方案一：

Muda(浪费)

方案二：

Mura(不均衡)

方案三：

Muri(超负荷)

最佳方案：

精益思想

考点：精益开发七浪费★

精益思想

希赛

考点：制造七浪费★

精益生产管理的主要关注点在浪费，而浪费管理方式是精益生产管理的核心。下述哪种情况是因为研发与制造之间花费时间过长而造成的缺陷？

A 库存过多

B 在计算生产流程中的总成本时出错

C 一名未培训过的员工在研发过程中拖延时间

D 制造产品所需的整个时间管理方式出错

精益思想

希赛

考点：制造七浪费★

精益生产管理的主要关注点在浪费，而浪费管理方式是精益生产管理的核心。下述哪种情况是因为研发与制造之间花费时间过长而造成的缺陷？

A 库存过多

B 在计算生产流程中的总成本时出错

C 一名未培训过的员工在研发过程中拖延时间

D 制造产品所需的整个时间管理方式出错

研发样品试产，特别是带有外壳模具的家电产品，在产线装机试产过程中，花费时间长，半成品多，产线整机测试不稳定，需要二次改良的产品库存。
初次试产的新产品，电路板与整机结构设计，会有电子元器件干涉，模具干涉等问题，没有合适可用的工装，产线装机费工时，产线测试老化程序不匹配，需嵌入式开发工程师修改匹配的软件。——家电厂研发部门测试岗同学分享

精益思想

考点：价值流图★ ★

价值流程图（Value Stream Mapping，VSM）用于分析信息或者材料的流动，从初始到结束，用来识别浪费的环节。识别出的环节即成为流程改进的地方。价值流程图的目的是为了辨识和减少生产过程中的浪费。

步骤一：选定需要分析的产品族

步骤二：调查现状

步骤三：绘制现状图

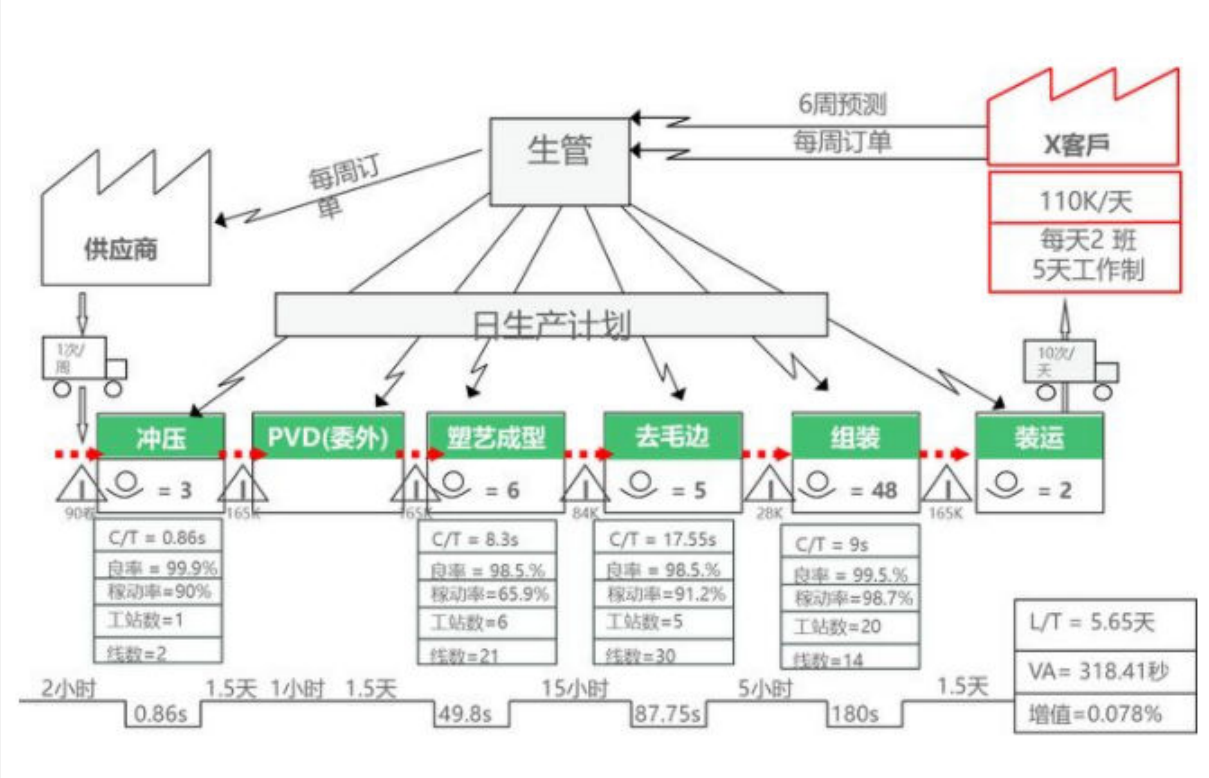
步骤四：检讨问题点

步骤五：绘制未来图

步骤六：制定改善计划

精益思想

考点：价值流图★ ★



步骤一：选定需要分析的产品族

步骤二：调查现状

步骤三：绘制现状图

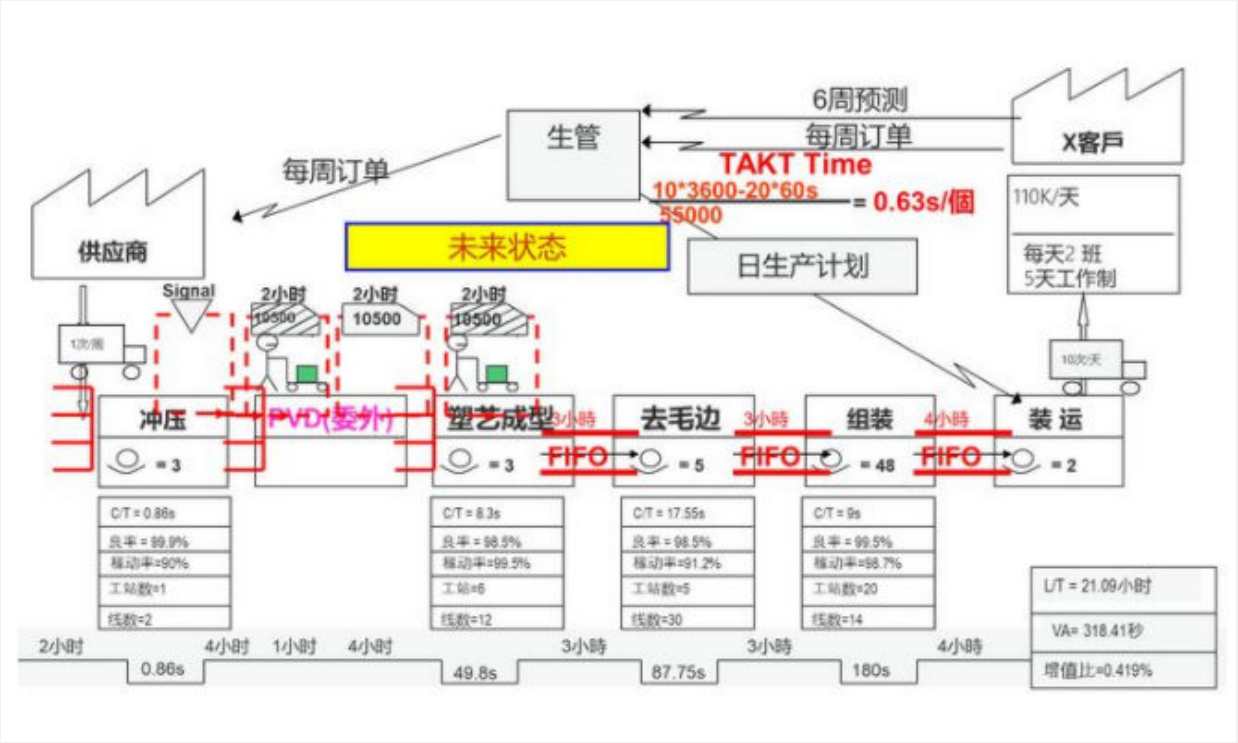
步骤四：检讨问题点

步骤五：绘制未来图

步骤六：制定改善计划

精益思想

考点：价值流图★ ★



步骤一：选定需要分析的产品族

步骤二：调查现状

步骤三：绘制现状图

步骤四：检讨问题点

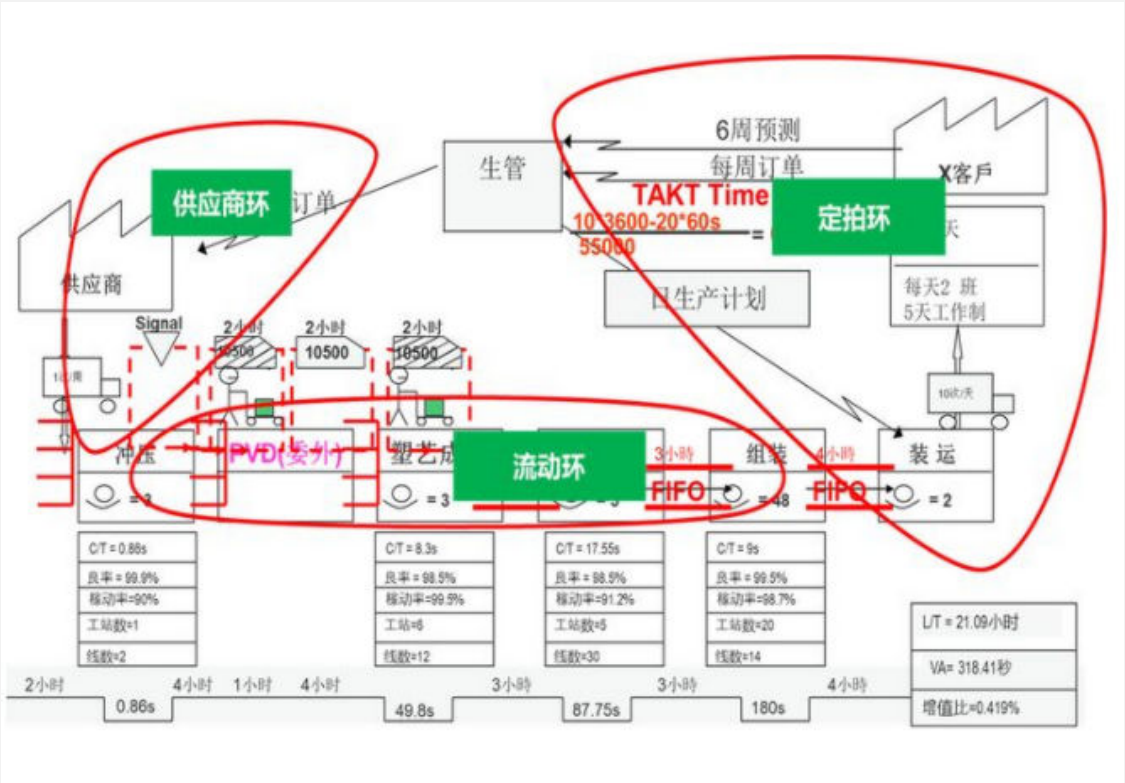
步骤五：绘制未来图

步骤六：制定改善计划

精益思想

希赛

考点：价值流图★ ★



步骤一：选定需要分析的产品族

步骤二：调查现状

步骤三：绘制现状图

步骤四：检讨问题点

步骤五：绘制未来图

步骤六：制定改善计划

精益思想

希赛

考点：价值流图★

团队需要交付一个关键的用户故事来解决一个问题。开始工作后，一名团队成员识别到对另一个供应商的依赖关系有两个星期的等待时间。这个等待时间将影响当前sprint的用户故事交付。该团队成员应该怎么做？

A 立即开始与团队合作，确定方案，加快速度并改进流程，按时交付用户故事

B 在回顾会议上与团队合作，确定方案，加快速度并改进流程，在下一次sprint交付用户故事

C 提醒该供应商的客户经理，让他们可以准备下一次会议

D 遵循流程，并在下一次会议上与供应商的客户经理讨论解决方案

精益思想

希赛

考点：价值流图★

团队需要交付一个关键的用户故事来解决一个问题。开始工作后，一名团队成员识别到对另一个供应商的依赖关系有两个星期的等待时间。这个等待时间将影响当前sprint的用户故事交付。该团队成员应该怎么做？

A 立即开始与团队合作，确定方案，加快速度并改进流程，按时交付用户故事

B 在回顾会议上与团队合作，确定方案，加快速度并改进流程，在下一次sprint交付用户故事

C 提醒该供应商的客户经理，让他们可以准备下一次会议

D 遵循流程，并在下一次会议上与供应商的客户经理讨论解决方案

精益-其他实践

希赛

- 自动化：创建自动化方法，只要检测到问题就停止生产。
- 看板：发出信号的卡片
- 拉动系统：流程中每个工序在零部件消耗完成时指示上游工序它需要更多库存
- 根本原因分析：找出某种情形发生的“深层”原因
- 改善：持续改进
- 安灯
- 单件流
- 现场现物
- 及时制

希赛

看板实践（Kanban）

看板Kanban

源于“准时制”库存控制和补给系统。最初用于杂货店，商店会根据货架不足情况来补给货架商品。大野耐一开发看板，1953年用于丰田。

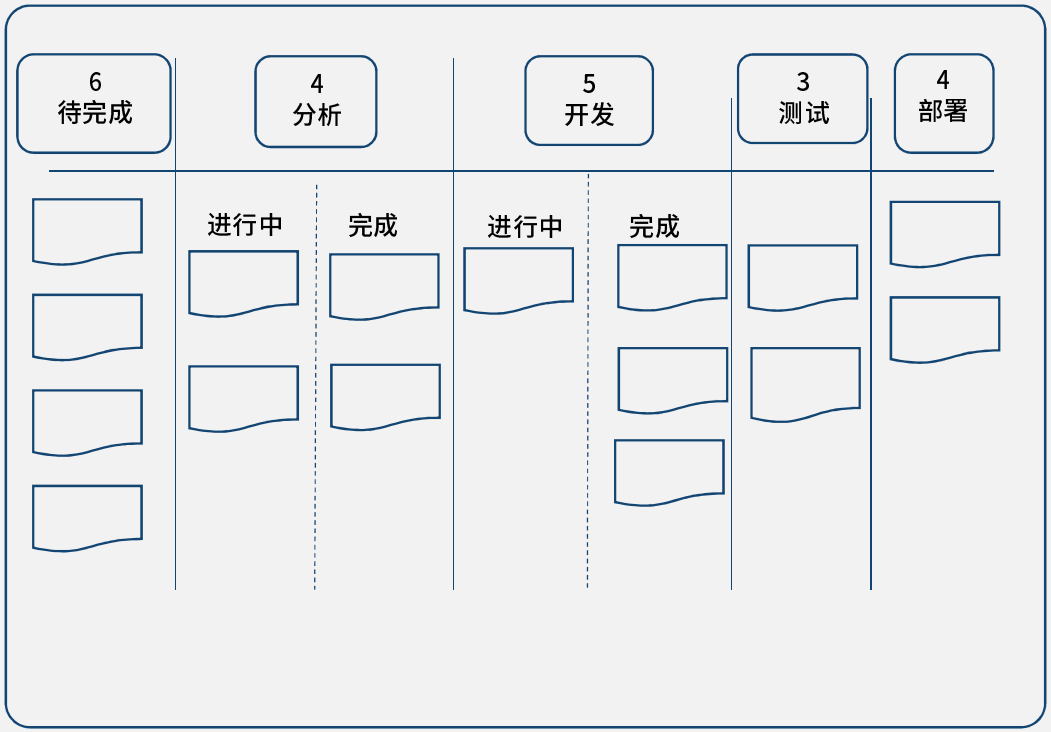
“看板”为“视觉符号”或“卡”。该信息发射源包含许多列：要完成的、进行中和已完成等，表示工作流的状态

看板方法适用于多种场合，可以确保工作流和价值交付的持续性。实施起来对原有管理方式破坏性较小，是原始的“原地出发”方法

未规定使用时间盒迭代，遵循在整个过程中持续拉取单个条目并限制在制品WIP的原则

看板Kanban

希赛



- 看板法源于精益；
- 整体性组织增量演变过程和系统变更框架，采用“拉式系统”来完成在制品；
- 看板面板利用列进入和退出策略以及限制在制品等制约因素，可提供一目了然的工作流、瓶颈、阻碍和整体状态信息；
- 限制在制品 (WIP, Work in Process)，让整个系统中的每份工作“完成”。

看板实践（Kanban）

希赛

狭义：

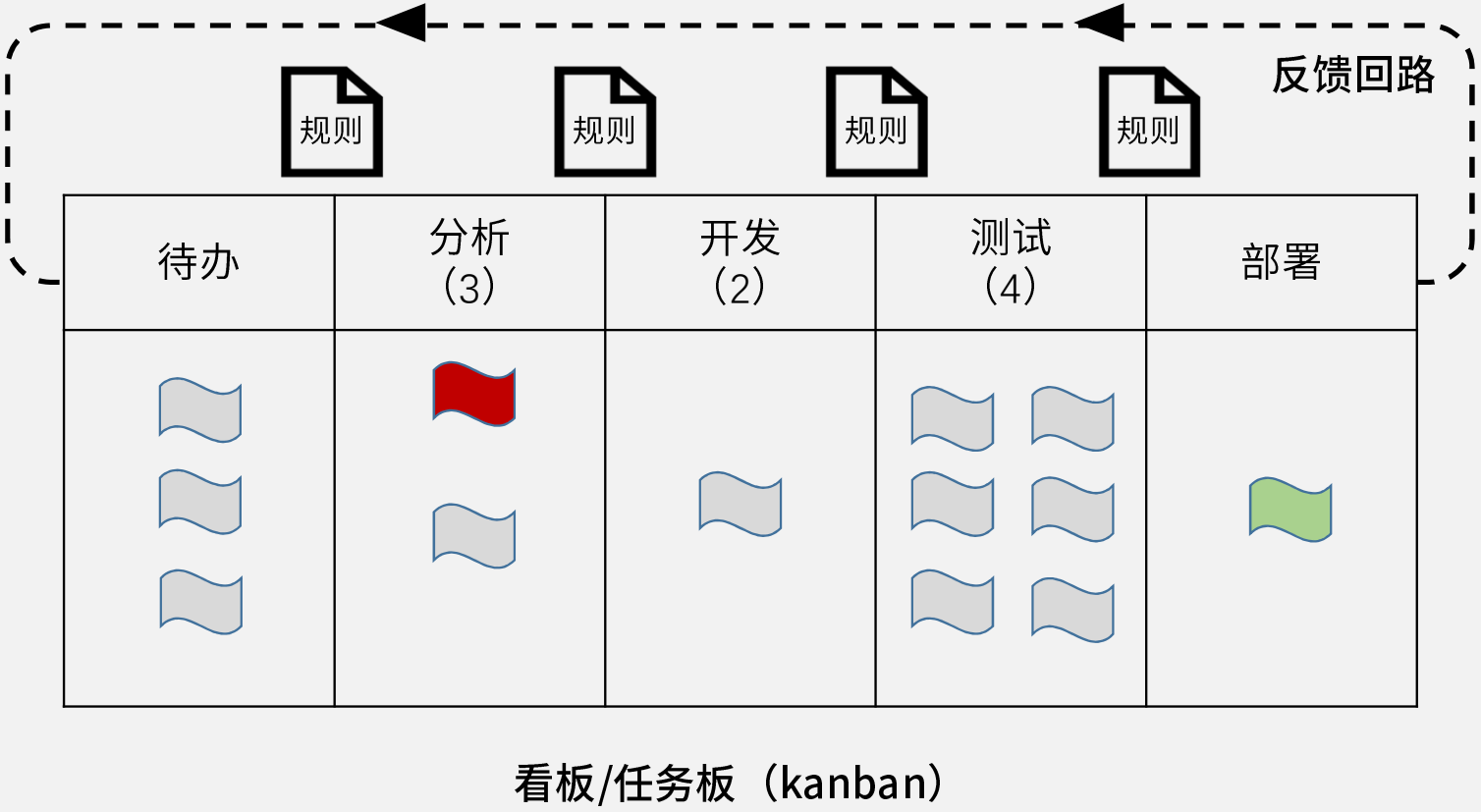
一块用来信息共享的板子

广义：

一种基于信息可视化的系统性实践方法

看板实践（Kanban）

希赛



看板/任务板（kanban）

看板六大核心实践

1、可视化工作流程

2、限制在制品（WIP）

3、管理流动（拉动）

4、显示化规则

5、建立反馈环路

6、协作式改进

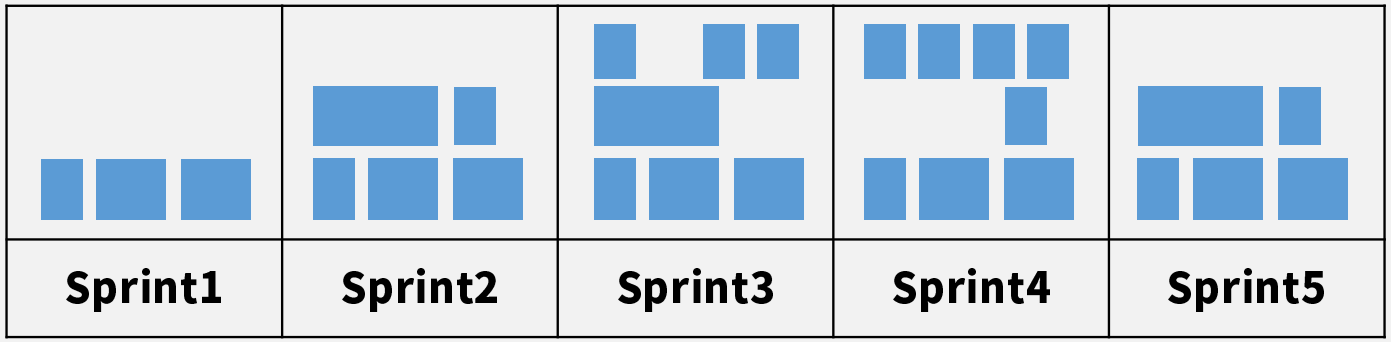
看板实践（Kanban）

希赛

可视化工作流	• 创建一个可视化的工作流示意图，方便跟踪每个工作项的当前状态
限制WIP	• 限制每个环节步骤中的工作项目数，使工作任务能均衡流动
管理流动	• 管理工作流使之快速而毫无中断的流动起来
显示化规则	• 明确定义和沟通团队所遵循的价值项流转规则，价值项从一个阶段进入下一个阶段比心达到的标准。
建立反馈环路	• 从你的流程中获得的反馈
协作式改进	• 应用可视化、限制WIP、价值流度量，暴露产品开发中的问题和瓶颈。回顾、分析、改进，并鼓励实验，推动整个团队持续改进。

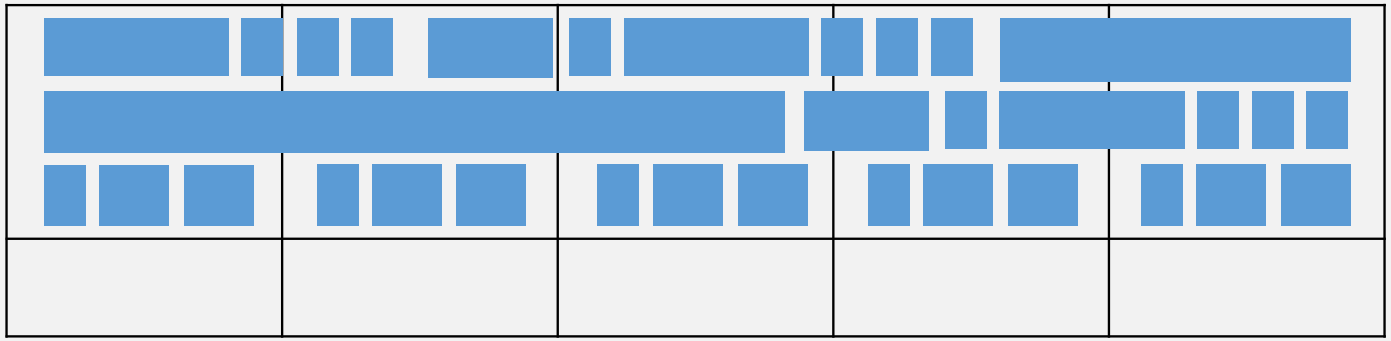
看板实践（Kanban）

希赛



Scrum

限制迭代周期



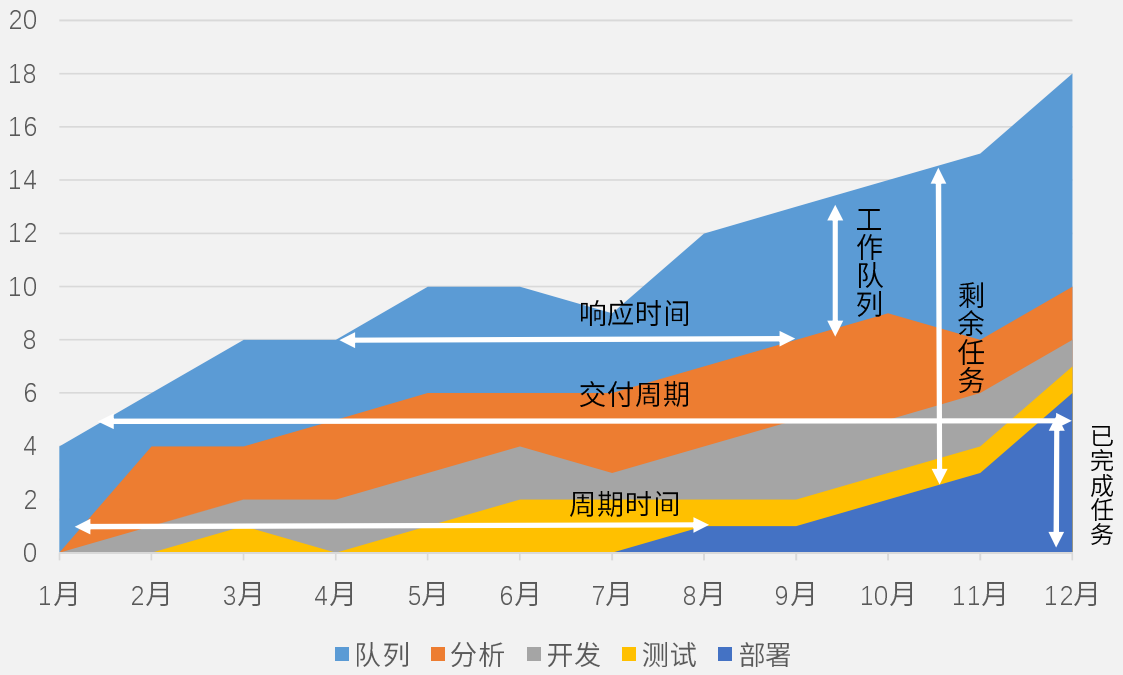
Kanban

限制在制品数量

看板实践（Kanban）

希赛

考点：信息发射器★



累积流图

（燃起图）

看板实践（Kanban）

希赛

考点：看板实践

一名关键干系人对项目的健康情况感到困惑。若要在将来防止这个问题，应该做什么？

A 仅提供口头项目报告

B 仅在规划会上与项目干系人讨论项目健康情况

C 创建符合公司报告标准的每周报告

D 可视化显示进度测量指标，并告知所有关键项目干系人其意义

看板实践（Kanban）

希赛

考点：看板实践

【要点】信息发射源（看板、累积流图、燃尽图）有助于与相关方了解项目，共识信息，管理风险等。★★

一名关键干系人对项目的健康情况感到困惑。若要在将来防止这个问题，应该做什么？

A 仅提供口头项目报告

B 仅在规划会上与项目干系人讨论项目健康情况

C 创建符合公司报告标准的每周报告

D 可视化显示进度测量指标，并告知所有关键项目干系人其意义

看板实践（Kanban）

希赛

考点：看板实践

Julie在一个使用看板改善流程的团队中。每天他们把索引卡片放在一个白板上，来显示流程各个状态中有多少个特性。接下来，他们把白板上每一列中的特性数累加起来，并创建一个面积图显示一段时间的特性总数。他们在使用什么工具？

A 累积流图

B 任务板

C 燃尽图

D 燃烧图

看板实践（Kanban）

希赛

考点：看板实践

Julie在一个使用看板改善流程的团队中。每天他们把索引卡片放在一个白板上，来显示流程各个状态中有多少个特性。接下来，他们把白板上每一列中的特性数累加起来，并创建一个面积图显示一段时间的特性总数。他们在使用什么工具？

A 累积流图

B 任务板

C 燃尽图

D 燃烧图

看板实践（Kanban）

希赛

考点：看板实践

你是一个集中办公的软件团队的Scrum教练，团队有5个人。在最近的回顾会议上，团队里有人提出团队有些负荷过重，有太多正在进行的工作。你最好做什么？

A 让团队在sprint中完成工作

B 让客户的所有请求都要经过你，使得团队不会负荷过重

C 尝试设置WIP限制

D 以上所有

看板实践（Kanban）

希赛

考点：看板实践

你是一个集中办公的软件团队的Scrum教练，团队有5个人。在最近的回顾会议上，团队里有人提出团队有些负荷过重，有太多正在进行的工作。你最好做什么？

A 让团队在sprint中完成工作

B 让客户的所有请求都要经过你，使得团队不会负荷过重

C 尝试设置WIP限制

D 以上所有

希赛

极限编程XP

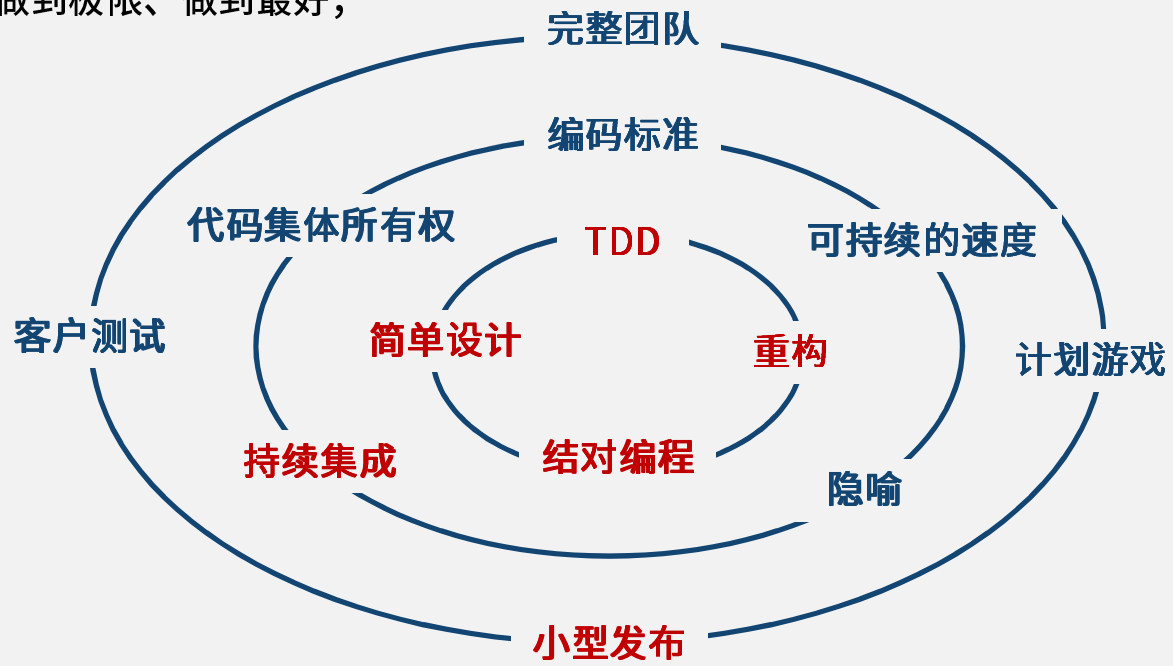
（eXtreme Programming）

极限编程XP

希赛

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。

“Extreme”（极限）是指，对比传统的项目开发方式，XP强调把它列出的每个方法和思想做到极限、做到最好；



极限编程XP

希赛

角色	XP教练，XP客户，XP程序员，XP跟踪员，XP测试员
核心 价值观	沟通，简洁，反馈，勇气，尊重
12个技术 实践	计划游戏，小版本，用户测试，集体代码所有权，编码标准，可持续的开发速度，比喻，持续集成，测试驱动，开发重构，简洁设计，结对编程。

极限编程XP-核心价值

希赛

- **简单（Simplicity）**：即实现最简单的解决方案。对当前需求进行设计、编码，不对将来可能的需求投入精力（因为可能变化）。
- **沟通（Communication）**：即追求有效的沟通。XP强调项目开发人员、设计人员、客户之间等有效地、及时地沟通，确保各种信息的畅通。
- **反馈（Feedback）**：即快速有效的反馈。收悉来自系统的反馈、来自客户的反馈、来自小组的反馈，通过反馈，良好沟通并修正系统。
- **勇气（Courage）**：即勇于重构和放弃。XP程序员要勇于对自己的代码进行重构，以及了解什么时候可以完全丢弃现有的某些代码。
- **尊重（Respect）**：即尊重团队成员和尊重工作。保证每次提交的代码是规范的，可以通过编译的。追求高质量。

极限编程XP-核心实践

希赛

- **完整团队(Whole Team)**，XP项目的所有参与者（开发人员、客户、测试人员等）一起工作在一个开放的场所中。
- **计划游戏(The Planning Game)**，计划是持续的、循序渐进的。
- **小型发布（Small Release）**，强调短周期内递增方式发布新版本。容易估算速度，控制工作量和风险，及时处理客户的反馈。
- **结对编程(Pair programming)**，两个程序员、并排坐在一起在同一台机器上构建代码。
- **测试驱动开发(Testing-Driven Development)**，编码前先编写单元测试用例，这是一个验证行为，更是一个设计行为。
- **重构(Refactoring)**，不影响功能的前提下对代码改进，保持代码尽可能的干净，使其更具有可维护性和可扩展性。
- **简单设计(Simple Design)**，团队保持设计恰好和当前的系统功能相匹配。

极限编程XP-核心实践

希赛

- **代码集体所有权(Collective Code Ownership)**，所有人对于全部代码负责，而非相互推诿责任。
- **持续集成(Continuous Integration)**，团队总是使系统完整地被集成。尽早暴露并消除由于重构、集体代码所有制所引入的错误。
- **客户测试(Customer Tests)**，作为定义功能需求的一部分，客户会描述测试以展示软件如何工作的。
- **可持续的速度（40-hour Week），**加班最终会扼杀团队的积极性，最终导致项目失败。团队只有持久才有获胜的希望。
- **编码标准（Code Standards）**，建立统一的代码标准，来加强开发人员之间的沟通，减少开发过程中的文档。
- **隐喻（System Metaphor）**，借用些沟通双方都比较熟悉的事物来做类比，减轻理解的复杂度。

极限编程XP

希赛

XP实践领域	主要	次要
组织	• 集中办公 • 整个团队 • 信息灵通的工作场所	• 真实客户参与 • 团队连续性 • 可持续节奏
技术	• 结对编程 • 测试驱动开发 • 增量设计	• 共用代码/集体所有制 • 代码和测试文档 • 重构
规划	• 用户故事 • 每周周期 • 每季周期 • 松弛	• 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	• 10分钟构建 • 持续集成 • 测试优先	• 单代码库 • 增量部署 • 每日部署

极限编程XP

希赛

对比	Scrum	XP
迭代长度	一般2-4周	一般1-2周
迭代内需求变更	遵循迭代计划，不允许迭代内变更	如果需要变化规模与原始需求相似，且原始需求并未开始，可以进行替换
迭代内优先级	团队酌情安排	严格按照优先级顺序
工程实践	不涉及	TDD，自动测试，结对编程，简单设计，重构等

极限编程XP

希赛

你是个XP团队的成员，在计划下一个周循环。有一个数据库设计任务需要完成。你的一个团队成员是数据库设计方面的专家，说他是唯一可以做这个工作的团队成员。最好的做法是什么?

A 鼓励充分应用各种技能以提高生产力

B 鼓励团队使用结对编程

C 鼓励这个专家作为团队一个初级成员的导师

D 在下一个每日站会上提出这个问题

极限编程XP

希赛

【要点】结对编程可解决专家不足、能力培训、团队冲突等问题。★★★

你是个XP团队的成员，在计划下一个周循环。有一个数据库设计任务需要完成。你的一个团队成员是数据库设计方面的专家，说他是唯一可以做这个工作的团队成员。最好的做法是什么?

A 鼓励充分应用各种技能以提高生产力

B 鼓励团队使用结对编程

C 鼓励这个专家作为团队一个初级成员的导师

D 在下一个每日站会上提出这个问题

极限编程XP

希赛

在将新产品推向市场之前，团队需要所有的特性和功能在客户验收测试期间获得97%的合格率，若要实现这个目标，团队应该采取哪个步骤？

A 遵守计划驱动的测试过程

B 批准和调试代码需要四天

C 实施高预测性测试过程

D 使用测试驱动（TDD）的开发实践

极限编程XP

希赛

【要点】TDD，自动化测试，持续集成，重构

在将新产品推向市场之前，团队需要所有的特性和功能在客户验收测试期间获得97%的合格率，若要实现这个目标，团队应该采取哪个步骤？

A 遵守计划驱动的测试过程

B 批准和调试代码需要四天

C 实施高预测性测试过程

D 使用测试驱动（TDD）的开发实践

水晶方法Crystal

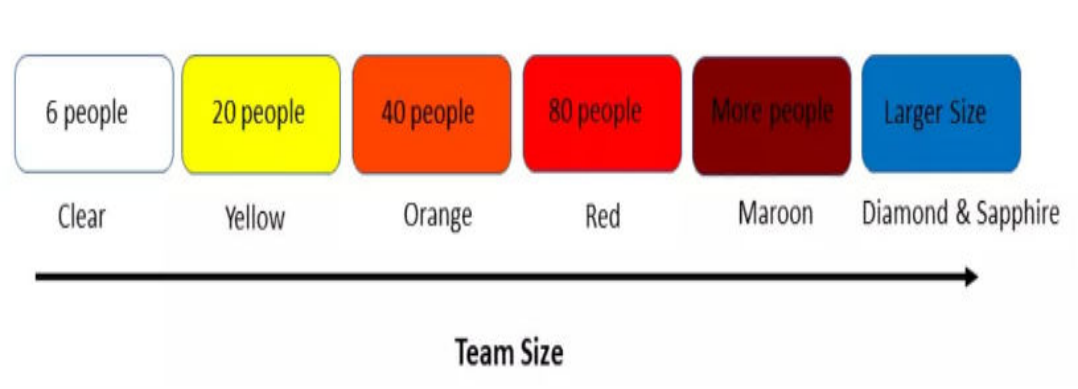
希赛

- 项目复杂程度越高，水晶颜色越深（从低到高依次是clear, yellow, orange, maroon, blue, violet）
- 源于宝石，不同面代表了根本的核心原则和价值观。
- 目前常用的：

  水晶橙色 1998年提出，针对C40,D40和E40项目

  水晶橙色web 2001年，在原基础上考虑更多的web开发

  水晶无色 2004年提出，针对C6和D6项目



ScrumBan

希赛

Scrumban 最初是 Scrum 到Kanban看板之间的过渡方法。
Scrum 作为框架，看板作为过程改进方法。

Scrumban将工作分解到许多小的“冲刺”，并利用看板面板来可视化和监督工作。故事列在看板面板上，团队通过使用在制品限制来管理其工作。

每日例会来维持团队之间的协作并消除阻碍。
Scrumban看板中没有预定义角色——团队保留其当前角色

特性驱动开发(FDD)

希赛

- 特性驱动开发Feature Drive Development, FDD是一种简单的构建产品或解决方案的强有力的方法。FDD的目的是满足大型软件开发项目的特定需求。

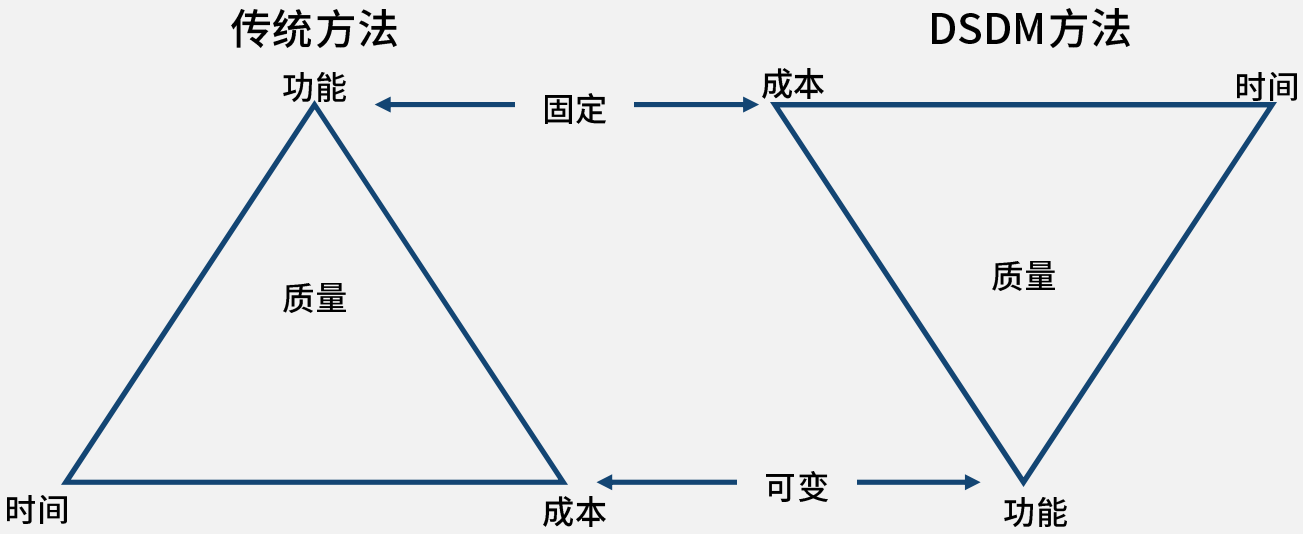
角色

- 项目经理
- 首席架构师
- 开发经理
- 首席编程人员
- 类负责人
- 领域专家

过程

- 开发整个模型
- 构建功能列表
- 依据功能规划
- 依据功能设计
- 依据功能构建

动态系统开发方法DSDM



- 动态系统开发方法DSDM是一种敏捷交付框架
- 最初目的是提高 20 世纪 90 年代普及的迭代方法的严格程度。
- 强调制约因素驱动交付而著称。
- 设置成本、质量和时间，利用范围优先级来满足制约因素要求。

敏捷扩展框架

希赛

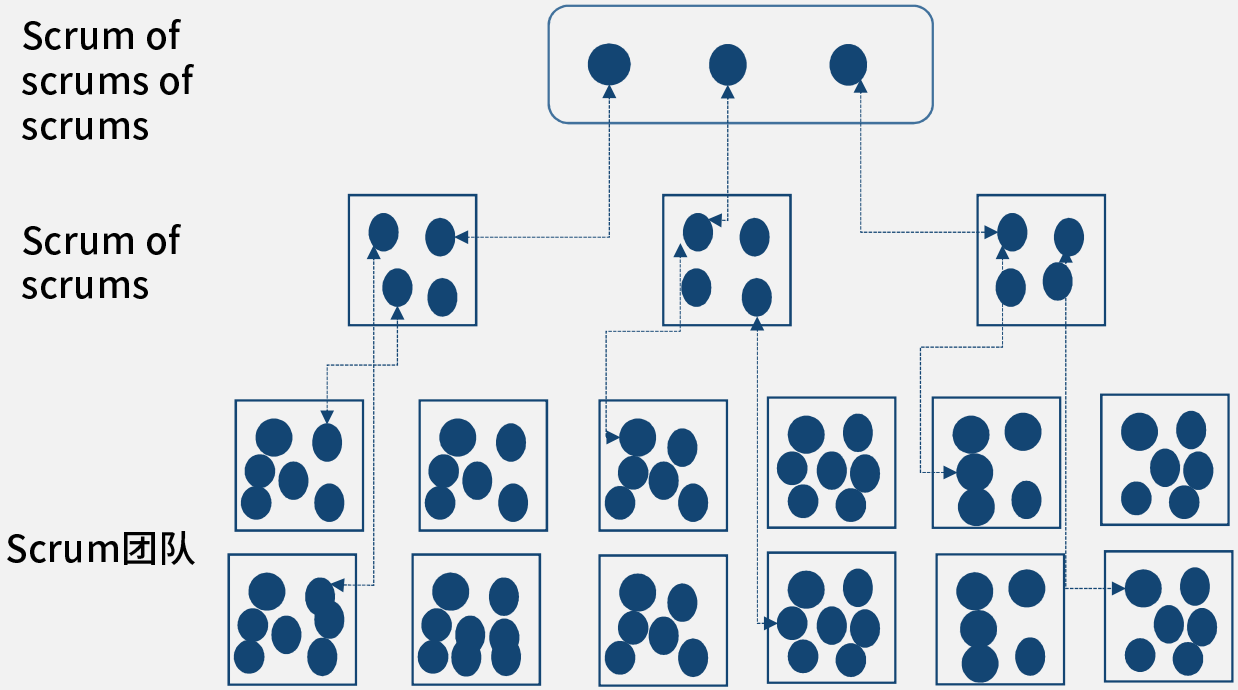
敏捷除了单团队单迭代的敏捷，还有一些敏捷扩展框架，适合多人团队实践敏捷方法。

Scrum of Scrums、大规模敏捷框架LeSS、大规模敏捷框架 SAFe、企业SCRUM和规范敏捷DA等，都是大规模敏捷

Scrum of Scrums，它是指一种多个团队围绕同一产品实施大规模敏捷开发工作的技术。他们需要协调讨论其相互依赖关系重点是如何整合软件的交付，在重叠的领域尤为如此。

Large-Scale Scrum (LeSS) 大规模敏捷开发，是一种产品开发框架。它根据扩展指导方针扩大敏捷开发规模，同时保留原有的敏捷开发目的。

Scrum of Scrums



希赛

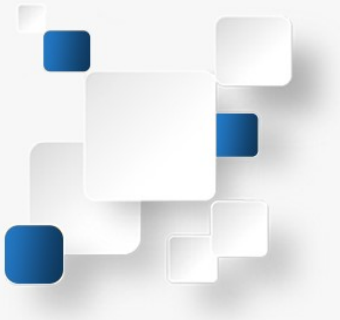
预测型与敏捷型
生命周期的一些比较

敏捷项目管理

希赛

PMBOK中关于在敏捷或适应型环境中需要考虑的因素

- 整合：自组织团队Self organizing team
- 范围：需求变化，多迭代版本，产品待办列表
- 进度：燃尽图、燃起图
- 成本：根据严格的预算控制范围，产品待办列表
- 质量：PDCA，频繁的增量交付，评审会议，回顾会议
- 资源：具有通才的自组织团队，共同承担责任
- 沟通：集中办公，面对面沟通，信息发射源，定期邀请相关方评审
- 风险：及时响应，作为用户故事添加，定期排列需求优先级
- 采购：大型项目，适应型针对某些可交付成果，变更只针对适应型工作
- 相关方：直接与客户对接，减少中间的层层管理级别；加快组织内部和组织之间的信息分享，敏捷提倡信息高度透明



感 谢 您 的 观 看