

PMP敏捷项目管理

《其他实践+敏捷变革》精华版

主讲：佩奇



敏捷04+05-其他敏捷实践+生命周期

极限编程	结对编程
	重构
	持续集成
	测试驱动开发TDD
	自动化测试
看板任务板	限制在制品WIP
	看板、精益
敏捷合同	动态系统开发方法DSDM ——— 敏捷三角
价值驱动交付	价值流程图
	累积流量图
其他概念	反向价值



极限编程XP

(Extreme Programming)



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



极限编程XP



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



核心实践：

1. **结对编程**：对于极限编程中的结对编程实践，反馈时间是最短的，一个人写代码，另一个在现场看，编码错误或对需求的误解会立刻被发现并给予反馈。
2. **持续集成**：多个团队及时将变更的代码集成到应用程序中。同时，在集成的过程中，可以快速发现一些编译的问题。对于可工作的软件来说，编译越频繁，发现问题的粒度会越小。
3. **驱动测试开发TDD**：倡导在开发之前先通过一个轻量级的测试来获得反馈。测试驱动开发倡导的理念是要在开发之前编写测试代码，因为开发人员在实际开发功能之前要清楚该功能应如何被测试。
4. **重构**：出于简化设计或提升代码可维护性的考虑，开发人员会在不改变代码功能的基础上，通过消除冗余等行为来对代码进行改良，这一过程被称为“重构”。
5. **自动化测试**：借助于测试工具、测试规范，从而局部或全部代替人工进行测试及提高测试效率的过程。



极限编程XP



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， <u>不允许</u> 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， <u>可以进行置换</u>
迭代内优先级	团队自行安排	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

XP提倡的方法：

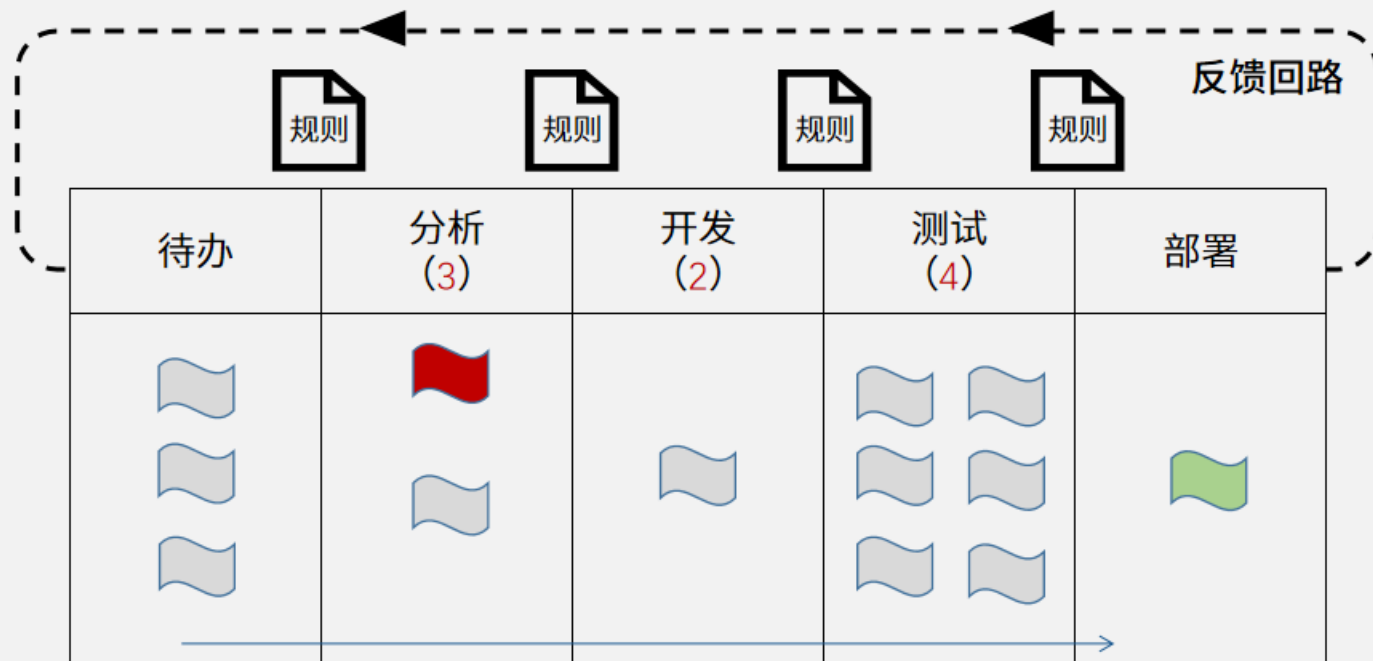
- 1、过程中至少有一名客户代表；
- 2、快速交付
- 3、结对编程
- 4、驱动测试开发TDD (Test-Driven Development)
- 5、代码集体所有
- 6、不加班（周不超过40h）
- 7、开放的工作空间
- 8、及时调整计划
- 9、重构



看板实践 (Kanban)



看板实践 (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、度量和管理工作流 (拉动)
- 4、显示化规则
- 5、建立反馈环
- 6、在协作及实验中改进

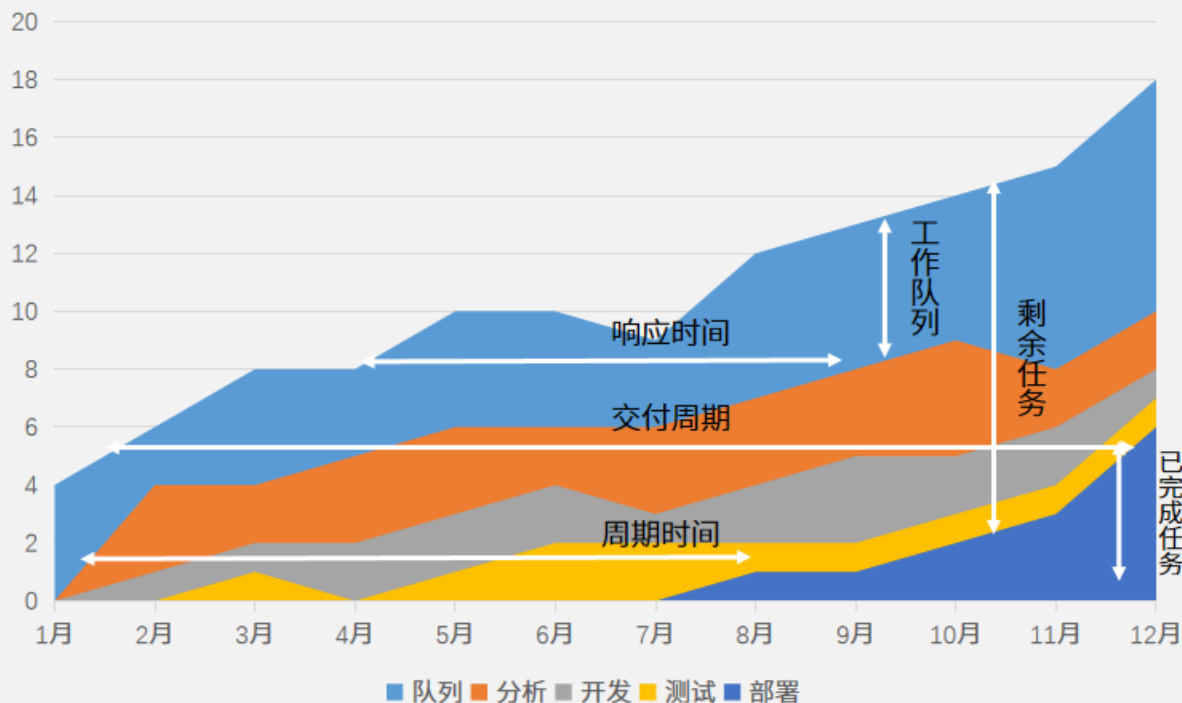
限制在制品：明确限制流程中各状态上同时进行的任务数上限。保持进行中工作的数量可以减少问题、瓶颈的发生，聚焦并持续改进。



看板实践 (Kanban)



考点：累计流量图★



累积流量图
(燃起图)



精益 (Lean)



精益思想



精益

精益思想就是**花最少的钱，办最好的事**，核心是**消除浪费**。

1. 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。
2. 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。
3. 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）。
4. 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。
5. 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。
6. **晚做决策**：尽早计划和最晚决策。
7. 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



考点：价值流程图★ ★

- 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。
- 价值流程图的目的目的是为了辨识和减少生产过程中的浪费。

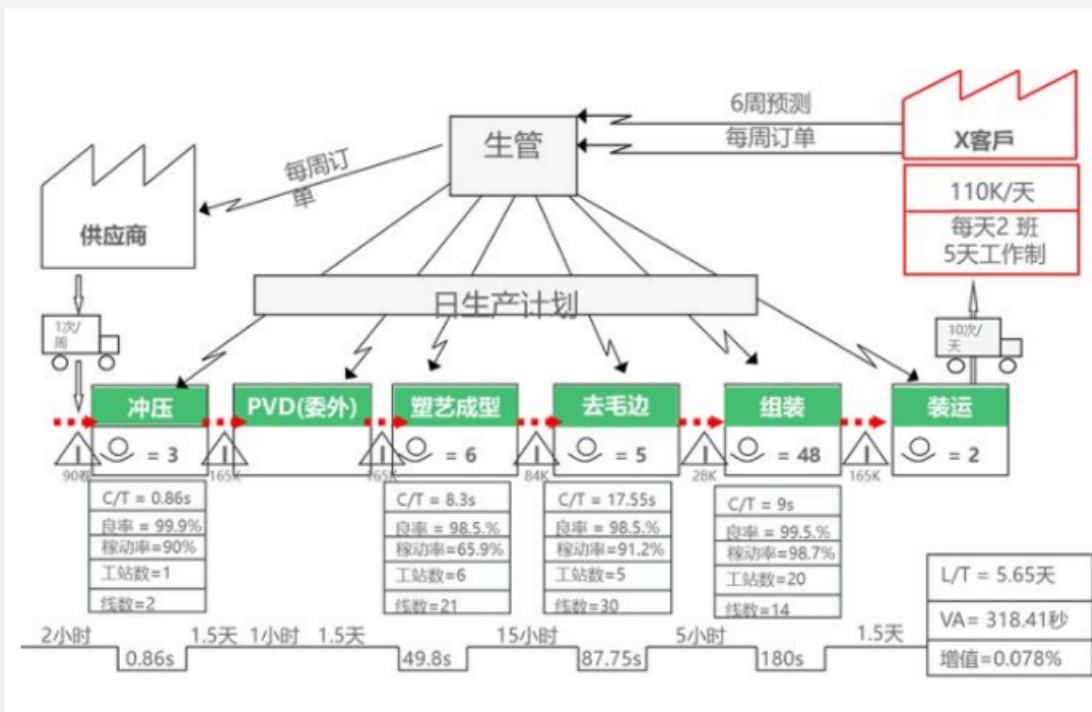




精益思想



考点：价值流程图★ ★



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



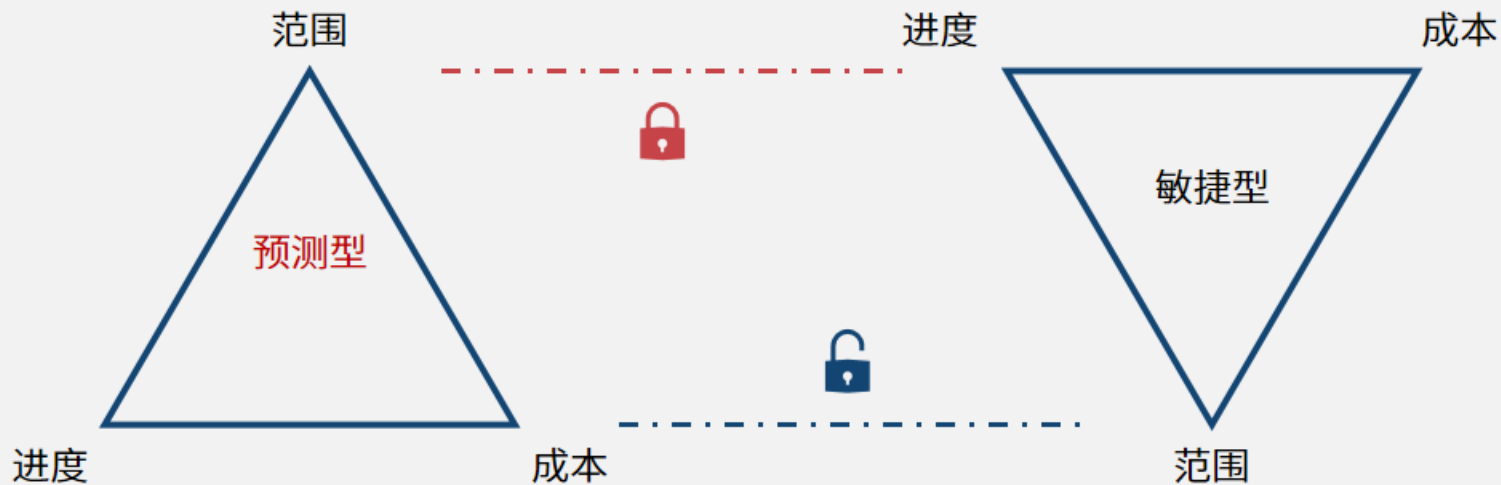
敏捷合同/协议



敏捷变革



考点：动态系统开发方法（DSDM）





敏捷变革



考点：敏捷采购与合同

多层结构	主协议：固定项目（担保、仲裁等）锁定； 主要服务协议：可能会变更的服务价格、产品说明等； 轻量级工作说明说：范围、进度和预算等更多动态变化项目。
总价增量	项目可以将范围分解为 总价微型可交付成果 （例如用户故事），而不是将整个项目范围和预算锁定到单个协议中。
固定时间和材料	将整体预算限制为固定数量。如客户需纳入新观点，用新的 工作替代原有工作 。
累进的时间和材料	共担财务风险法，合同期限之前交付，则奖励；反之，则惩罚。



敏捷概念



考点：反向价值

- 反向价值：就是能为项目带来损失的那部分称之为反向价值，相对于的就是完成客户的需求能够产生价值。比如一个风险会导致项目亏损4000元，那这个反向价值就是-4000。



混淆概念

考点：易混淆概念系列

- 抖音号：loveclass888
- 定期更新pmp考试的易错概念





感谢您的观看