

## 希赛 PMP 敏捷知识点集锦

敏捷宣言 .....	4
敏捷十二原则 .....	4
敏捷阶段框架 .....	6
四种生命周期及对比 .....	6
仆人式领导 .....	8
敏捷团队特征 .....	8
敏捷团队的角色 .....	9
项目经理作为仆人式领导的职责 .....	9
卡诺分析与 MoSCoW .....	11
MoSCoW .....	12
DoD 工作完成准则 .....	13
铁三角与敏捷三角 .....	14
用户故事 .....	14
故事点估算之宽带德尔菲、计划扑克 .....	15
速度 .....	17
信息发射源 .....	17
看板与任务板 .....	17
燃尽图、燃起图、限制累积流量图 .....	19
问题发现与解决 .....	22
五问法 .....	22
鱼骨图 .....	22

控制图.....	23
在制品及限制在制品.....	23
冲突管理.....	23
Scrum 的 33355.....	25
三个支柱.....	25
三个角色.....	26
三个工件.....	27
产品待办事项列表 Product backlog.....	27
冲刺待办事项列表 Sprint Backlog.....	27
可交付产品增量 Increment.....	28
五个事件.....	28
冲刺 Sprint.....	28
Sprint0 和 SprintH.....	28
冲刺计划会议 Sprint Planning.....	29
每日站会 Daily Scrum.....	30
迭代评审会议 Sprint Review.....	31
迭代回顾会议 Sprint Retrospective.....	31
五大价值观.....	32
敏捷章程.....	32
团队章程.....	32
持续改进.....	33
敏捷合同.....	33
风险管理.....	34



更多备考资料及考试咨询可加 V: educity1426

---

风险刺探、架构刺探.....	35
XP 极限编程.....	35
XP 的技术实践.....	36
精益看板.....	37
最小可行产品与最小可售功能.....	40

希赛教育



更多备考资料及考试咨询可加 V: educity1426

【特别提示：需要理解敏捷宣言、敏捷原则和 SCRUM 方法，其他类容大部分可以通过以上三大块得到】

### 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：

个体和交互 胜于 流程和工具

工作的软件 胜于 详尽的文档

客户合作 胜于 合同谈判

响应变化 胜于 遵循计划

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

### 敏捷十二原则

准则 1：我们的最高目标是，通过尽早和持续地交付有价值的软件来满足客户

准则 2：欢迎对需求提出变更，即使在项目开发后期也不例外。

敏捷过程要善于利用需求变更，帮助客户获得竞争优势。

准则 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。

准则 4: 项目实施过程中, 业务人员与开发人员必须始终通力合作。

准则 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。

准则 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是面对面的交谈。

准则 7: 可用的软件是衡量进度的首要衡量指标。

准则 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保持步调。

准则 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

准则 10: 简洁, 即尽最大可能减少不必要的工作。这是一门艺术

准则 11: 最佳的架构、需求和设计出自于自组织团队。

准则 12: 团队要定期反省怎么做才能更有效, 并相应地调整团队的行为。

## 敏捷阶段框架

相较于结构化项目管理的启动、规划、执行、监控、收尾五大过程组，部分流派将敏捷也做了一个五阶段框架：构想、推演、探索、适应、结束。他们不能直接跟结构化项目管理中五大过程组完全匹配对应，但是可以粗略这样对应。

构想阶段，得到产品愿景；推演阶段，得到用户故事清单、产品待办事项列表、产品发布计划等；探索阶段，执行得到完成的用户故事；适应阶段对完成的进行把关；结束阶段，完结整个项目。

## 四种生命周期及对比

**预测型生命周期：**一种更为传统的方法，提前进行大量的计划工作，然后一次性执行；执行是一个连续的过程。也称为瀑布式。

**迭代型生命周期：**这种方法允许对未完成的工作进行反馈，从而改进和修改该工作。

**增量型生命周期：**这种方法向客户提供各个已完成的，可能立即使用的可交付成果。

敏捷生命周期：这种方法既有迭代，也有增量，便于完善工作，频繁交付。

方法	需求	活动	交付	目标
预测型	固定	整个项目仅一次执行	一次交付	管理成本
迭代型	动态	反复执行直至修正	一次交付	解决方案的正确
增量型	动态	对给定增量执行一次	频繁更小规模交付	速度
敏捷型	动态	反复执行直至修正	频繁小规模交付	通过频繁小规模交付和反馈实现 的客户价值

计划始终贯穿其中，每种生命周期都有计划要素，不同之处在于完成多少计划以及何时完成

## 仆人式领导

特征：建立社区、倾听、同理心、成长承诺、治愈能力、管理能力、意识、洞察力、说服力、化繁为简能力。

- 1、帮助、促进团队发展。
- 2、为团队消除障碍。
- 3、为他人贡献铺路。
- 4、教育相关方，使其了解为什么要敏捷以及如何敏捷。
- 5、通过指导、鼓励和帮助为团队提供支持。
- 6、通过技术项目管理活动，如量化风险分析来帮助团队。

7、庆祝团队的成功，为团队与外部团队合作提供支持，并起到桥梁作用。

## 敏捷团队特征

- 1、团队规模：3-9 名。
- 2、理想情况下，集中办公。
- 3、100% 专职成员、跨职能团队、通才型专家、自我激励自律的成员。



4、鼓励自我管理团队，由团队成员决定谁执行下一阶段工作。

5、敏捷团队与仆人式领导一起茁壮成长，领导支持团队的工作方法。

6、团队频繁创造产品增量，团队集体对工作负责并共同拥有完成工作所需的所有必要技能。

7、限制在制品（WIP）、彼此依赖实现交付。

8、以各种方式开展合作（结对、群集、群体开发）。

### 敏捷团队的角色

1、跨职能团队成员：拥有各种必要技能；以常规节奏交付潜在可发布产品；核心职责是在短时间内交付任务。

2、产品负责人：指导产品的开发方向；创建、维护产品待办事项；根据商业价值排序任务；提供反馈。

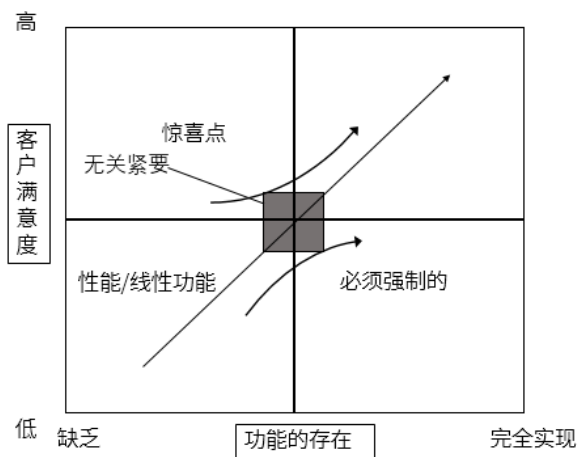
3、团队促进者：确保遵循敏捷流程；引导、指导、消除障碍。

### 项目经理作为仆人式领导的职责

1、帮助、促进团队发展。

- 2、为团队消除障碍。
- 3、为他人贡献铺路。
- 4、教育相关方，使其了解为什么要敏捷以及如何敏捷。
- 5、通过指导、鼓励和帮助为团队提供支持。
- 6、通过技术项目管理活动，如量化风险分析来帮助团队。
- 7、庆祝团队成功，为团队与外部团队合作提供支持，并起到桥梁作用。

## 卡诺分析与 MoSCoW



Kano 用于给需求进行优先级排序，它将需求分为以下四种类型：

**作为阈值的功能：**必须的功能。是产品要成功就必须具备的那些功能。改善增加阈值功能数量到一定程度，对客户满意度提升就没有多少影响了。



更多备考资料及考试咨询可加 V: educity1426

**线性功能：**“越多越好”的功能。它的改善与数量增加能直接提升用户满意度，它的降低和数量减少也直接降低用户满意度。

**兴奋点和惊喜点：**指提供了很高满意度，并常常为产品增加额外价格的那些功能。但是缺少它不会让客户满意度降到中性以下。

**无关紧要属性：**这些特性无论从哪方面来说对客户都没有影响。因为客户对其根本就不关心，我们应该努力消除、最小化或者延迟交付这些特性。

#### MoSCoW

莫斯科法则，就是 must or should, could or would not, 用于给需求进行优先级排序。

按照“Must: 必须做的; Shoud: 应该做的; Could: 可以做的; Would not: 不要做的”来做，保证 PO 所需要的 Must、Should 完成，并力争 Could 能完成；在响应变更时，优先考虑牺牲 Could 乃至 Should 需求特性。

## DoD 工作完成准则

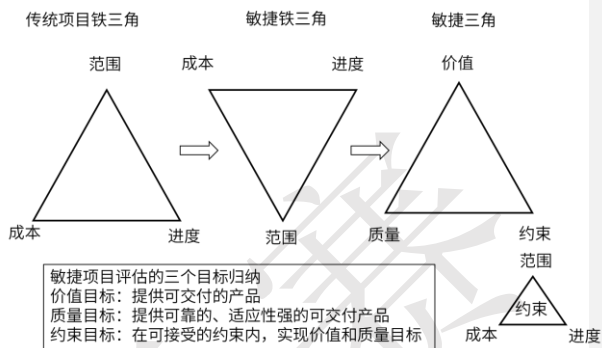
Definition of Done，工作完成准则。是帮助干系人对项目工作达成一致的必要准则，由团队所有成员一起决定，一般在敏捷各个层次的计划上制定，包括如下三个层级：

发布 DoD，如：完成发布规划范围内的那些需求；至少通过一次发布的回归测试；修复所有等级为 1、2 的缺陷……

迭代 DoD，如：所有完成的用户故事已验收；所有代码得到静态分析，纠正最高级别的不符合项；所有新增代码得到评审；所有完成的用户故事都有对应的测试用例……

用户故事 DoD，如：用户故事最终的描述符合 INVEST；用户故事都有对应的测试用例……

## 铁三角与敏捷三角



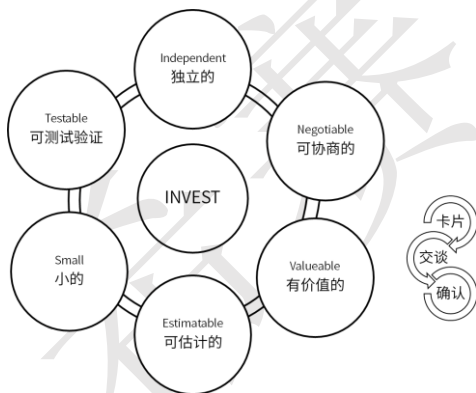
传统项目铁三角：范围相对固定，成本进度是变量；敏捷铁三角，成本进度相对固定，范围是变量；敏捷三角强调价值和质量管理，约束（范围、进度、成本）。

## 用户故事

用户故事描述了对用户、系统或软件购买者有价值的功能，用来收集客户需求。在卡片正面描述用户的需求，在背面描述该功能对应的期望：如估值、交付实践、验收标准等。正面描述实例如下：

用户故事卡片	
作为一个	{角色}
我想要	{潜在的需求}
所以	{商业价值}

另外用户故事需满足 INVEST 属性，并遵循 3C 原则。



### 故事点估算之宽带德尔菲、计划扑克

用户故事的故事点估算，是通过规模的相对大小来排列故事，最终基于这些排列来进行估计。常见估算有宽带德尔菲和计划扑克、亲和估算等。

宽带德尔菲，一种基于团队共同参与的估算方法，一群专家匿名提交估算结果，彼此不知道真实的结果，这样可以提升团队成员对结果的认同感，也可以避免产生一些“光环效应”。一般会进行多轮，直到达成共识。相较于一般的德尔菲，宽带德尔菲是专家匿名投票后，会开放空间让大家交流和讨论。

计划扑克，每人 10 张数字牌，每个人选一张卡片，代表这个故事所花费的成本，这个时间点选择的卡片不能给他人看。所有参与者同时展示自己的卡片。团队一起来讨论这些估计值，尤其对异常值（最高的和最低的）要着重讨论。

亲和估算，快速估计大规模需求未完项的一种技术，利用衬衫尺寸（尺码 S/M/L/XL/XXL 等）、咖啡杯尺寸（小杯、中杯、大杯等）或斐波那契序列（序列中的数字等于前两个数字值之和）中的数字将用户故事快速置于规模类似的群组中。



## 速度

一个 Sprint 中完成的故事点数。这个数按时间取平均值，来预测多个 Sprint 中可以完成的工作量。随着团队成员彼此熟悉，以及成员对项目工作的不断熟悉，速度值会趋于稳定。它反应了团队历史工作能力。但是注意速度是独特的，在不同团队、不同项目间进行比较没有意义。

## 信息发射源

信息发射源是一系列高可视化展示信息的方式。包括燃尽图、燃起图、累计流量图、故事地图、任务板、看板等等。它有高度可见、透明、实时、简单等特点。

## 看板与任务板

看板来自于精益，它显示所有的特性的状态，帮助团队了解工作在流程中是如何流动的。保持在特性层次而不是任务板的任务层次。它不能直观准确地帮助了解团队成员在完成哪个任务，但能帮助直观查看工作流程中各阶段正在进行的工作数量。

任务板用来透明展示任务的状态，可以跟踪进度，让团队对项目的工作任务状态清楚明白，帮助团队成员自我组织。

两者相同之处体现在：任务板、看板都是信息发射源，都能将项目信息直观的展示给团队成员。

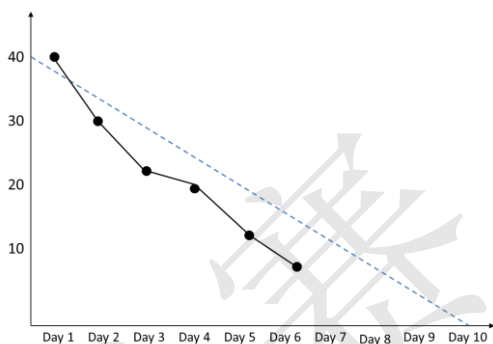
不同之处：

1、看板是为了帮助团队跟踪工作在流程中如何流动，能发现流程中的瓶颈，同时也能帮助了解特性的进展状态。工作流程与团队工作方式一致。

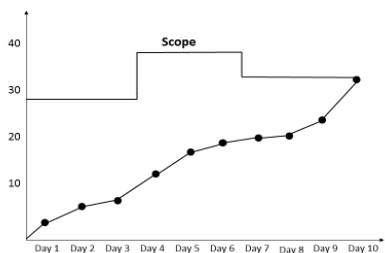
2、看板中的卡片显示的是特性，有可能没有细化到任务层次，不是跟踪团队成员的具体工作任务状态的最佳信息发射源。

3、任务板显示所有任务的状态，能让团队成员清楚透明的跟踪进度，并进行自适应调整计划。

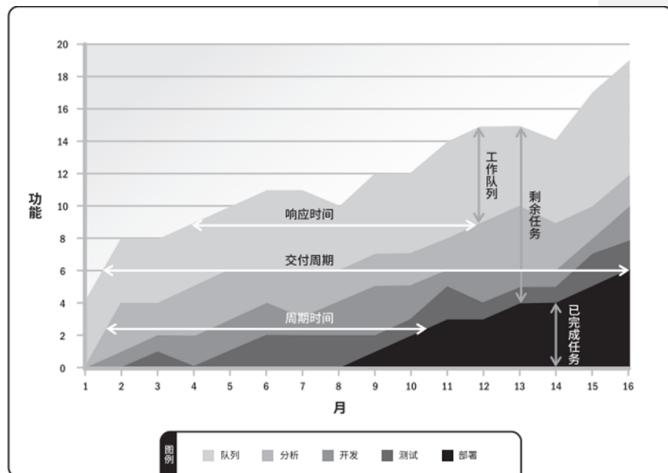
## 燃尽图、燃起图、限制累积流量图



燃尽图以图形化的方式展现了团队本迭代内剩余的工作量（Y轴）与工作日（X轴）的关系，因整体上总是递减而得名。其目的是架空迭代度或项目的进度，追踪剩余总和并预测达到迭代目标的可能性。

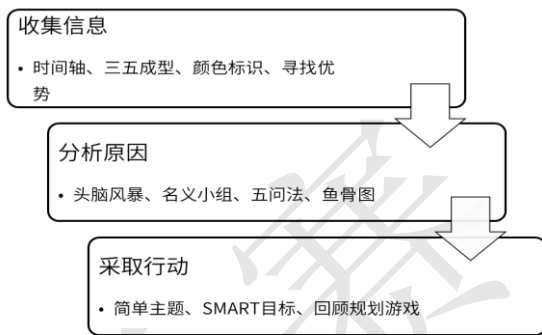


燃起图以图形化方式展示了项目或团队多个迭代的需求累积完成情况（Y 轴）与迭代时间（X 轴）的关系，因整体上总是递增而得名。它可以反映出范围的调整变化。



累积流量图展示了项目已完成的任务和剩余的任务，还可以通过不同颜色区域边界的水平距离来确定响应时间、交付周期和周期时间。能够直观反映跟踪各个流程阶段的任务进度，还能帮助发现工作中的瓶颈。

## 问题发现与解决



敏捷问题检测技术有鱼骨图、5Whys、控制图和限制在制品。

### 五问法

五个为什么是一种通过不断重复询问为什么来识别问题或缺陷的根本原因分析技术。每个为什么回答成为识别下一个为什么的驱动力。

### 鱼骨图

又称石川图或因果图，是一种有效识别问题或缺陷的根本原因分析方法，与五问法结合使用。

## 控制图

控制图是一个设置有控制界限的图表，它被用来监控一段时间内的行为的过程。控制界限有助于反映项目是否在控制中。如果信号超出界限意味着过程失控，如果连续 7 个点都处于中线同一侧也意味着失控。

## 在制品及限制在制品

在制品指的是团队已经开始进行但还没有完成的需求。在制品数量越多则前置时间越长，且质量越低。但是在制品过少，又会导致人员、生产资源的利用率降低。因此需要平衡限制在制品数量，达到整体工作流程的优化。

## 冲突管理

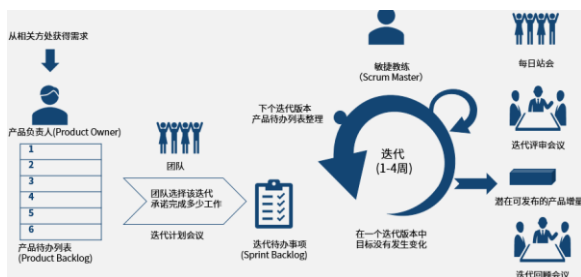
冲突等级	特征方式	管理方式
一级，解决问题	团队意识到有问题待解决，团队保持关注发生什么改变及如何弥补	合作，寻求双赢
二级，不同	成员有异议和冲突，进入冷战	授权团队去解决

意	模式	问题
三级，竞争	冲突变成竞争。很多问题未能解决，团队关注是赢，而不是解决问题。	谈判，调停
四级，讨伐	冲突演变为运动。光解决冲突还不够。团队成员坚信在这个问题上的“另一边”不会改变。他们可能相信，唯一的选择就是让对方从团队中退出或者他们自己从团队中退出。	重塑安全的环境，使用外交手段，干预、协调缓和和问题
五级，战争	冲突成为战争。没有建设性成果	强制停止任何伤害

批注 [y1]: 根据老师意见所作修改。



## Scrum 的 33355



### 三个支柱

**透明：**过程中的关键环节对相关人是显而易见的，同时保证干系人对这些关键环节理解是统一的。

**检查：**Scrum 使用者必须经常检视 Scrum 的工件和完成 Sprint 目标的进展，检视频率应适宜。

**适应：**如果检视发现一个或多个方面偏离可接受范围以外，并且将会导致产品不可接受时，就必须对过程或过程化的内容加以调整。调整工作必须尽快执行如此才能最小进一步的偏离。

Scrum 规定了 4 个仪式，用于检视和适应。

### 三个角色

产品负责人 (Product Owner)：主要负责确定产品的功能和达到要求的标准，维护产品代办事项列表，指定软件的交付的内容，确保开发团队对产品待办事项列表有足够深的了解，同时有权利接受或拒绝开发团队的工作成果。改变产品待办事项优先级都必须经过产品负责人。

团队负责人 (Scrum Master)：主要负责整个 Scrum 流程在项目中的顺利实施和进行，以及清除挡在客户和开发工作之间的沟通障碍，使得客户可以直接驱动开发。

开发团队(Scrum Team)：主要负责软件产品在 Scrum 规定流程下进行开发工作，人数控制在 3~9 人左右 (PO, SM 不包含在人数中，除非参加执行冲刺列表中的工作)，团队获得授权，自组织和管理他们的工作。每个成员可能负责不同的技术方面，但要求每个成员必须要有很强的自我管理能力和责任心，属于整个开发团队。为团队提供了一种一起成功，一起失败，一起调整，一起改进的途径。

## 三个工件

### 产品待办事项列表 Product backlog

它是一份涵盖产品中已知所需每项内容的有序列表，它是产品需求变动的唯一来源，它永远不是完整的。包括描述、次序、估算和价值，包括测试描述，在完成时证明其完整性。

### 冲刺待办事项列表 Sprint Backlog

它是一组为当前冲刺 Sprint 选出的产品待列表项。Sprint 待办列表是拥有足够细节的计划，任何进度的变化可以在每日 Scrum 站会中清晰地看到。

要注意的是：

一、开发团队在 Sprint 期间修改 Sprint 待办列表，使得它在迭代期间涌现。只要开发团队在按计划开展工作的过程中，学习到了关于达成 Sprint 目标所必需的工作，Sprint 待办事项列表就会发生涌现。新工作将被放入 Sprint 待办列表中，剩余工作量被估算及更新，失去意义的工作将被移除。

二、开发团队对 Sprint 待办列表全权负责。

批注 [y2]: 与老师协商后进行的修改。

## 可交付产品增量 Increment

增量是一个冲刺 Sprint 完成的所有产品待办表项的总和，以及之前所有 Sprint 所产生的的增量的价值总和。无论产品负责人是否决定发布它，增量必须可用。

### 五个事件

#### 冲刺 Sprint

冲刺，也翻译成迭代，在 Scrum 中冲刺是类似橄榄球比赛中的冲刺：大家团结一致，为了完成该 Sprint 的目标疯狂向前冲。一个 Scrum 项目会划分为多轮冲刺，每一轮冲刺都有一些需要完成的目标，也包含了冲刺计划会议、每日站会、冲刺评审会议和冲刺回顾会议以及这一轮冲刺所需完成的工作和工作完成的结果。

#### Sprint0 和 SprintH

Sprint0 是第一个 Sprint 之前一个特殊周期，也称“Inception”阶段。一般在四周以内，完成典型准备工作：团队组建、培训、确定 PB、第一个迭代故事、架构设计、原型开发、定义项目各类 DoD、办公场地准备、初始发布计划。

SprintH 特殊迭代，也称为“Release”迭代。用于对已完成迭代的增量的补充，这个迭代不发布新的需求。

### 冲刺计划会议 Sprint Planning

Sprint 中要做的工作在 Sprint 计划会议中来做计划。这份工作计划是由整个 Scrum 团队共同协作完成的。

Sprint 计划会议是有时间盒限定的，以一个月的 Sprint 来说最长为 8 小时。对于较短的 Sprint，会议时间通常会缩短。Scrum Master 要确保会议顺利举行，并且每个参会者都理解会议的目的。Scrum Master 要教导 Scrum 团队遵守时间盒的规则。

Sprint 计划会议回答以下问题：

- 接下来的 Sprint 交付的增量中要包含什么内容？
- 要如何完成交付增量所需的工作？

话题一：这次 Sprint 能做什么？

话题二：如何完成所选的工作？



## 每日站会 Daily Scrum

每日 Scrum 站会为接下来 1 天的工作制定计划，通过检视上次站会依赖的工作及预测将要到来的工作来优化团队的协作和性能，它是进行检视与适应的一个关键会议。成员回答了以下三个问题：

昨天，我完成了什么？今天我准备做什么？是否有什么问题？

注意事项：

- 1、它的时间盒为 15 分钟内，SM 需要教导团队遵循。
- 2、它是内部会议，有外部参与者，要确保他们不干扰会议。
- 3、它不是状态汇报会议，也不是问题解决会议。相关议题的详细讨论要在站会后独立开展。

## 迭代评审会议 Sprint Review

在 Sprint 快结束时举行，用以检视所交付的产品增量并按需调整产品待办列表。在 Sprint 评审会议中，Scrum 团队和利益攸关者协同讨论在这次 Sprint 中所完成的工作。根据完成情况和 Sprint 期间产品待办列表的变化，所有参会人员协同讨论接下来可能要做的事情来优化价值。这是一个非正式会议，并不是一个进度汇报会议，演示增量的目的是为了获取反馈并促进合作。一个月的冲刺不超过 4 小时，参会者为团队与 PO、相关系人。核心在于交付、讨论“价值”。

## 迭代回顾会议 Sprint Retrospective

是 Scrum 团队检视自身并创建下一个 Sprint 改进计划的机会。回顾会议发生在 Sprint 评审会议结束之后，下个 Sprint 计划会议之前。对于长度为一个月的 Sprint 来说，回顾会议时间最长不超过 3 小时。对于较短的 Sprint 来说，会议时间通常会缩短。Scrum Master 要确保会议举行，并且每个参会者都明白会议的目的。Scrum Master 确保会议是积极的和富有成效的。Scrum Master 教导大家遵守时间盒的规则。Scrum Master 对 Scrum 过程负责，作为团队的一员参加该会议。是团队内部会议，PO 可根据情况参加或不参加。核心在于“改进”。

## 五大价值观

承诺、专注、开放、尊重、勇气。

承诺（Commitment）愿意对目标做出承诺；专注（Focus）全身心都用到你承诺的工作上去；开放（Openness）团队内所有信息对所有人开放；尊重（Respect）每个人都有他独特的价值和经验；勇气（Courage）勇于承诺，履行承诺，敢于说不。

## 敏捷章程

我们为什么要做这个项目？愿景。

谁会从中受益？如何受益？目标

达到哪些条件才意味着项目完成？

我们将怎样合作？这说明预期的工作流。

## 团队章程

团队社会契约

团队价值观

工作协议

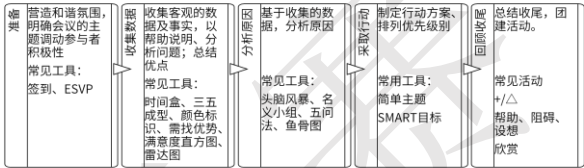
基本规则



团队规范

持续改进

回顾会议步骤



敏捷合同

敏捷合同的签署，更遵循“客户协作高于合同谈判”，强调基于价值交付的合作，签署时可从如下方面增加敏捷性：多层次合同设计；采用尽可能短的、增量的、迭代的、多批次的交付计价形式；范围动态调整；指定专职人员；分化风险等。

固定总价合同可以在合同中约定范围调整条款；而采用工料合同、成本补偿合同可以为项目提供更好的敏捷性。

## 风险管理

敏捷风险管理与传统项目管理生命周期相似，分为规划风险（按敏捷价值观，团队决策风险管理工作基本准则）、识别风险、响应风险（规划、实施风险应对）、风险评审（回顾会议中进行）四个阶段。

### 风险应对策略：

1、规避：如果整体项目风险有严重的负面影响，并超出商定的项目风险临界值，采用该方法使项目不受风险影响。如减小范围，采用熟悉供应商，增加资源等。

2、减轻：降低不利风险事件发生的概率和/或影响，到可接受的临界点。

3、转移：将风险应对的责任和后果转移给第三方承担，不消除风险。

4、接受：可采用主动策略，建立应急储备来应对；也可采用被动策略，团队发生时再处理。

注意的是，题干中的“通常”，它指的是敏捷方法中通常采用定性，但是敏捷方法不排斥定量，如团队成员一致认可定量分析的必要性，则可以将定量分析的工作纳入计划中。

## 风险刺探、架构刺探

架构刺探用来验证一个技术方法是否可行。风险刺探用于排除项目可能存在的风险。

它们都是采用一个简短的实践盒对不确定性进行尝试。它们都会为迭代中增加用户故事。

## XP 极限编程

### 价值观

勇气：有勇气接受挑战，为项目挺身而出。

尊重：团队伙伴相互尊重，团队中的每个人都信任彼此能完成任务。

沟通：极限编程强调沟通，采用建立信息化场所（洞穴和公共区域布局），信息发射源，渗透式沟通来增加沟通的效率。

反馈：通过迭代、持续集成、审查、单元测试等促进反馈，从而达到不断的改善。

简单：尽可能减少不必要的工作，这是一门艺术。

### 迭代时间盒

季度循环及周迭代，团队会为每个迭代增加松弛，提高迭代调整的灵活性。

### Scrum 与极限的区别

Scrum 关注项目管理和产品开发，主要用来使团队更有组织。极限编程主要还是迭代和增量的应用，缺乏 scrum 的结构性和严格性，如没有 scrum 中对会议时间盒的具体要求。同时，极限编程采用了更多的技术实践提高项目的敏捷性。

### XP 的技术实践

**计划游戏：**快速制定计划，随着细节的不断变化而完善。

**小版本：**系统的设计要能够尽可能早地交付。

**用户测试：**要求至少有一名实际的客户代表在整个项目开发周期在现场负责确定需求、回答团队问题以及编写功能验收测试。

**集体代码所有制：**每个成员都有更改代码的权利，所有的人对于全部代码负责。

**编码标准：**强调通过指定严格的代码规范来进行沟通，尽可能减少不必要的文档。

**可持续开发速度：**要求项目团队人员每周工作时间不能超过 40 小时，加班不得连续超过两周，从而保证开发的速度的稳定。

**隐喻：**通过隐喻来描述系统如何运作、新的功能以何种方式加入到系统。它通常包含了一些可以参照和比较的类和设计模式。

**持续集成：**提倡在一天中集成系统多次，而且随着需求的变化，要不断的进行回归测试，避免了一次系统集成带来的困境。

**测试驱动：**先写测试代码再编写程序。

**重构：**代码重构是指在不改变系统行为的前提下，重新调整、优化系统的内部结构以减少复杂性、消除冗余、增加灵活性和提高性能。

**简洁设计：**任何时候都应当将系统设计的尽可能简单。不必要的复杂性一旦被发现就马上去掉。

**结对编程：**由两个程序员在同一台电脑上共同编写解决同一问题的代码。通常一个人负责写编码，而另一个负责保证代码的正确性与可读性。

## 精益看板

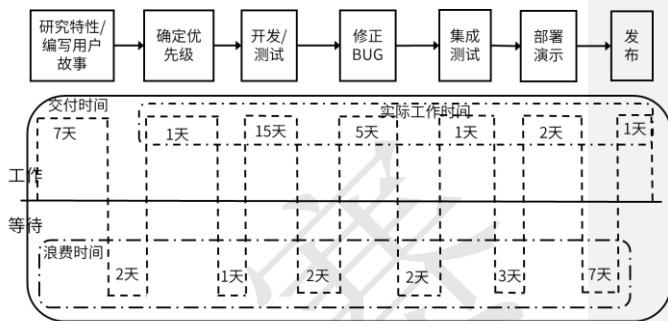
### 精益的七大核心概念

消除浪费、增强学习、尽晚决定、尽快交付、授权团队、建构完整性、综观全局。

### 常见的浪费

必须消除的浪费的三大来源	7 种制造浪费	7 种软件浪费
浪费、不均衡、负荷过重	库存、运输、动作、等待、缺陷、过度生产、过度加工	部分完成的工作、多余的流程、多余的特性、任务切换、等待、动作、缺陷

### 价值流程图



流动效率=实际工作时间/交付时间X100%= 25/32=78%

### 精益需了解的思维工具

**查找浪费:** 找出浪费的根本原因, 从而消除浪费。流程中的浪费可以通过价值流程图工具**辨识和减少**。

**批注 [y3]:** 此处根据网络查询补充。

**排队论:** 它是一种数学中运筹学的一个分支。精益中使用排队论来确保成员不会工作负荷过重, 从而保证持续、稳定的节奏。

**拉动系统:** 可以理解为团队成员完成一个任务, 再选择一个新任务。保证了工作的专注, 减少并行。

**选项思维：**将工作计划及工作任务当作选项，每日站会上，团队可以在任务板上自由地增加和去除任务，它们没有截止日期，强调的是尽其所能的完成产品工作，而非做出过度承诺。

**延迟成本：**了解某个任务延迟带来的反价值，可以帮助团队决定哪些任务先完成。

**感知完整性/概念完整性：**感知完整性指考虑一个特性是否能很好地满足用户的需求。概念完整性指考虑特性是否能很好地合作来形成一个统一的产品。

**基于集合的开发：**团队花时间准备一组选项，在最后责任时刻才决定采用哪个选项。

## 最小可行产品与最小可售功能

MVP 是最小可行性产品。由 Eric Ries 在《精益创业实战》中提出，指的是用最快、最简明的方式建立一个可用的、并可以推向市场的产品或演示，以满足用户核心需求，并获取反馈。

MMF 是一种小型的、自包含的特性，它可以快速开发并为用户提供重要的价值，是准备交付产品对应的最小功能集合，可以理解为 MMF 在交付之后就是 MVP。



MMF 是项目过程中对项目要交付客户需求的描述，而 MVP 是客户需求的实现或演示。