

# 敏捷知识点补充

## 一、敏捷思想理念总结

### 1、敏捷宣言：

我们一直在实践中探寻更好的软件开发方法，身体力行的同时也帮助他人，由此我们建立了如下价值观：

- 个体与互动 重于 流程和工具
- 工作的软件 重于 详尽的文档
- 客户协作 重于 合同谈判
- 响应变化 重于 遵循计划

也就是说，尽管右项有其价值，我们更重视左项的价值。

### 2、敏捷 12 原则：

- 我们最重要的目标，是通过持续不断地及早交付有价值的软件使客户满意
- 欣然面对需求变化，即使在开发后期也一样，为了客户的竞争优势，敏捷过程掌控变化
- 经常地交付可工作的软件，相隔几个星期或一两个月，倾向于采取较短的周期
- 业务人员和开发人员必须相互合作，项目中的每一天都不例外
- 激发个体的斗志，以他们为核心搭建项目，提供所需的环境和支援，辅以信任，从而达成目标
- 不论团队内外，传递信息效果最好效率也是最高的方式是面对面的交谈
- 可以工作的软件进度的首要度量标准
- 敏捷过程提倡可持续开发，责任人、开发人员和用户要能够共同维持其步调稳定延续
- 坚持不懈的追求技术卓越和良好设计，敏捷能力由此增强
- 以简洁为本，它是极力减少不必要工作量的艺术
- 最好的架构、需求和设计出自自组织团队
- 团队定期地反思如何能够提高成效，并依此调整自身的举止表现

### 3、敏捷核心实践和原则

包括但不限于以下：

- 在迭代中尽早演示交付的价值
- 用户故事反映商业价值和优先级
- 客户持续改进
- 所有需求的验收测试

- 回顾
- 可持续的节奏或速度
- 沟通
- 高度可视化

### 敏捷不包含什么

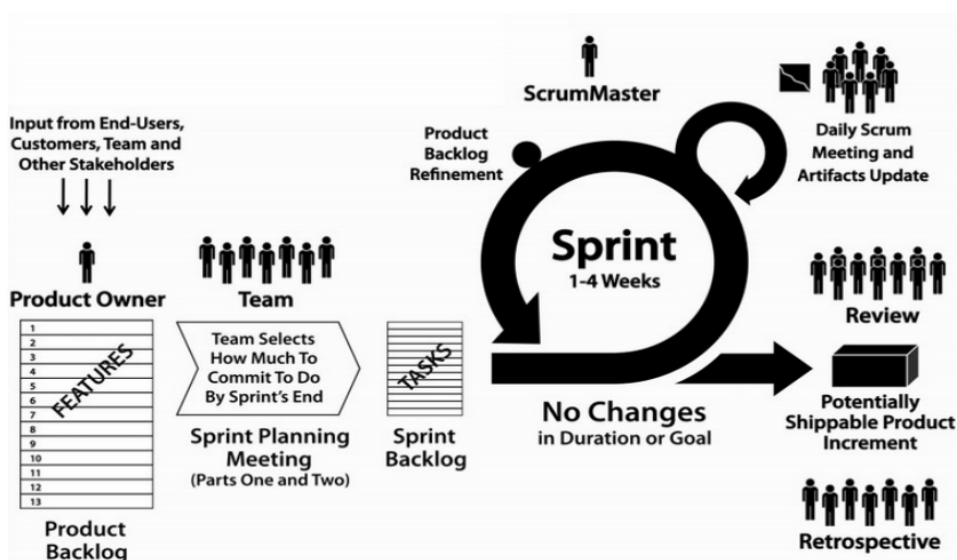
- 预先的设计和需求收集
- 项目完工预测
- 死亡行军项目：项目团队为弥补估算差距无偿加班
- 强迫使用工具，例如任务管理工具
- 自上而下管理和控制
- 大量文件，特别的状态报告，软件架构图，软件需求规格说明书，测试计划

### 敏捷益处

- 强调协同合作，团队授权，频繁过程演示
- 轻量级，依靠白板，概要卡片和便利性工具
- 吸引开发者的开发重点
- 更快的上市时间和高优先级特征驱动开发生命周期,
- 关注,拉而不是推
- 容易理解
- 满足客户的需要

## 二、敏捷里的“3355”-Scrum 角色 工件 活动

SCRUM 整体框架视图  
(3 种角色、 5 个仪式、 3 个工件、 5 个价值观)



(开放 Openness, 专注 Focus, 勇气 Courage, 承诺 Commitment, 尊重 Respect)

## 1、Scrum 概述

Scrum 是用于管理产品开发的单个团队过程框架。该框架包含 Scrum 角色、事件、工件和规则，采用迭代方法来交付工作产品。

### 1) Scrum 流行的原因：

- ☞ Scrum 提供简单和可证明的结果
- ☞ 它包含其他敏捷工程技术
- ☞ 它强调小型团队和团队授权
- ☞ 欢迎需求的变更
- ☞ 它允许单一来源的优先项目工作开展
- ☞ Scrum 会议包括日常状态会议
- ☞ 提供团队在冲刺阶段一个潜在的可交付增量承诺

### 2) Scrum 三大支柱

#### 透明性：

- ☞ 过程或项目的各个方面必须是对结果负责任的，透明的；
- ☞ 运用信息发射源,让这些关键信息,如产品待办事项列表,冲刺待办事项、障碍、风险和项目进展对所有的利益相关者是透明的。

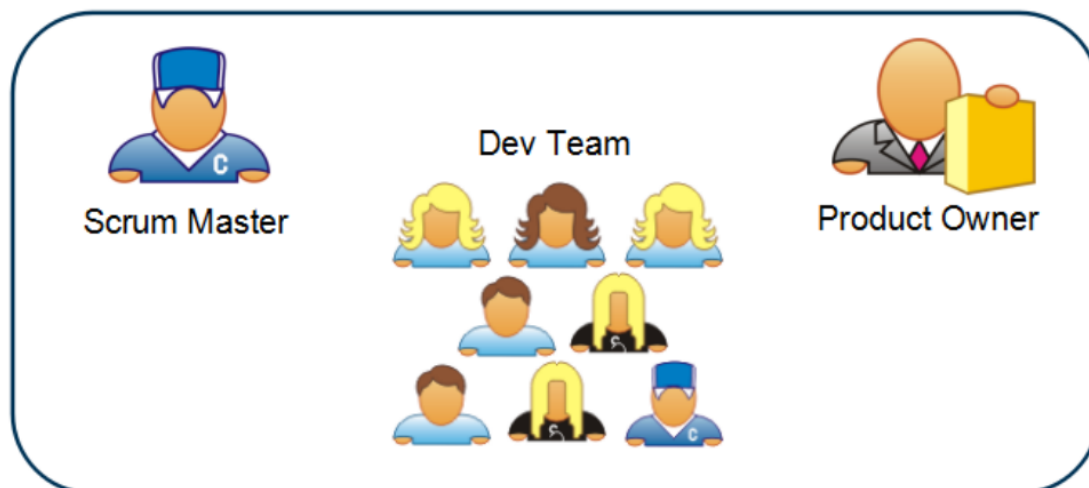
#### 检视：

- ☞ 团队根据项目目标定期检查他们的绩效和进展；
- ☞ 他们不断寻找问题和计划的偏离。

#### 调整：

- ☞ 基于观察期间的检查,采取必要的变更流程,以避免问题再次发生,提高项目交付成功率。

## 2、Scrum 角色



Scrum 团队由 **5 到 9 个 (7±2)** 团队成员组成。有三种类型角色:

- ☞ **产品负责人 (PO)**: 产品负责人定义项目愿景、需求和优先级, 对产品成功负责。
- ☞ **Scrum Master**: 负责团队, 并移除障碍, 帮助他们实现产品负责人所设定的目标。
- ☞ **开发团队**: 自组织、跨职能。他们协同工作, 以确定如何最好地满足产品负责人的目标。

团队中有“鸡”和“猪”的角色, “猪”的角色包括 scrum master, PO, team; “鸡”的角色是指团队成员以外的管理角色

### Scrum 角色的关键点

#### 1) 产品负责人:

- 清晰地表达产品待办列表项
- 对产品待办列表项进行排序, 最好地实现目标和使命
- 优化开发团队所执行工作的价值
- 确保产品待办列表对所有人可见、透明、清晰, 并且显示 Scrum 团队的下一步工作
- 确保开发团队对产品待办列表项有足够的理解

#### 2) Scrum Master:

Scrum Master 负责确保所有人都能正确地理解并实施 Scrum。因此, Scrum Master 要确保 Scrum 团队遵循 Scrum 的理论、实践和规则。

Scrum Master 是 Scrum 团队中的**服务型领导**。Scrum Master 帮助 Scrum 团队外的人员了解他们如何与 Scrum 团队交互是有益的, 通过改变他们与 Scrum 团队的互动方式来最大化 Scrum 团队所创造的价值。**Scrum Master** 在期望设定和管理中扮演重要角色, 以此去创建高绩效团队。

#### A) Scrum Master的职责是:

- ☞ 在项目生命周期早期定义基本规则;
- ☞ 确保团队理解干系人期望;
- ☞ 同团队沟通项目愿景, 有利于确保团队;
- ☞ 认识到他们的目标同项目总目标紧密一致;
- ☞ 以连贯的单元模式工作;
- ☞ 对愿景给予承诺。

#### B) Scrum Master制定的基本规则包括:

- ☞ 设定Scrum仪式的开始-结束时间;
- ☞ 保持对主题的专注减少分散;
- ☞ 会议期间杜绝中断;
- ☞ 允许团队成员特别是初级成员言论自由;
- ☞ 在制定决策前应广泛搜集所有成员意见。

#### 3) 团队:

有自主权选择如何最好地满足目标, 并且为之负责。

### 3、Scrum 三个工件

Scrum 的工件以不同的方式表现工作任务和价值, 可以用来提供透明性以及检视和调整的机会。Scrum 中的工件就是为了最大化关键信息的透明性, 因此每个人都需要有相同的理解。

#### ① 产品待办列表 (Product Backlog)

- ☞ 产品需求列表;
- ☞ 产品负责人对该列表进行优先级排序;
- ☞ 待办事项列表中的条目以用户故事的形式呈现;

#### ② Sprint 待办列表 (Sprint Backlog)

- ☞ 是产品待办列表的子表, 只记录当前迭代的工作;
- ☞ 将用户故事拆分成任务, 团队成员主动领取任务;
- ☞ 团队成员可以添加、删减或者更改迭代中的任务。

#### ③ 产品增量 (PSPI: Potentially Shippable Product Increment)

- ☞ 团队在迭代内完成交付成果, 集成到以往的迭代成果中, 形成增量式的交付。
- ☞ 每次交付的用户故事必须符合验收条件。

## 4、Scrum 会议(5 个仪式)

### ① 冲刺计划会议：

Scrum 团队的所有成员出席，在此次会议中，开发团队识别当前冲刺开发交付的产品待办事项中的故事。

这个会议时间箱为：一个月的冲刺，会议时间 8 小时，4 个小时用于选择故事和 4 个小时估算分配。

### ② 每日站立会议：

由 **Scrum Master** 和开发团队参加，产品负责人可以自行选择是否参加。每日站立会议是快速专注的会议，用来分享迭代或迭代进展。

每个团队成员就他们将要完成的任务对其他人做口头承诺。

每个团队成员回答以下问题：

- ☞ “昨天做什么？”
- ☞ “今天将做什么？”
- ☞ “遇到了什么问题？”

每日立会只有猪的角色可以发言，鸡的角色不可以发言

这次会议时间箱 15 分钟，每天发生在同一时间和地点。

### ③ 冲刺评审会议：（review）

这次会议是由 Scrum 团队的所有成员参加。

开发团队将可能移交的可交付物开发特性演示给干系人和项目发起人。

Sprint 评审会议的结果是一份修订的产品待办列表，确定很可能进入下个 Sprint 的产品待办列表项。

这个会议时间箱为一个月的迭代，4 个小时，比冲刺计划会议的持续时间更短。

1、冲刺评审是在迭代末期进行的时间盒（有指定时间限制）会议，此时不断变化的解决方案展示给利益相关者，他们的反馈得到收集。

该会议是：



- 针对冲刺末期召开;
- 被时间盒定义到四个小时, 按月冲刺和较短的时间段;
- 冲刺评审会议由包括开发团队, 产品负责人, **Scrum Master**, 和企业的利益相关者的整个团队出席;
- 这些冲刺评审会议被团队通过录音、快照来展示产品。

## 2、冲刺评审的益处

进行常规冲刺评审会议有助于:

- 产品根据利益相关者的需要在变化;
- 任何反馈或升级在即将到来的冲刺或发布中被记录和强调;
- 优先级排序的待办事项将被展示给利益相关者去评估是能够满足他们的期望;
- 逐步完善未来的项目计划。

## 3、冲刺评审的重要性

在一个 2 周冲刺的项目中, 没有组织冲刺会议将导致项目进度落后于整整一个月。

这是因为:

- 开发的需求没有满足利益相关者的期望;
- 为即将到来的冲刺所选择的需求, 没有同利益相关者的需求保持一致。

## ④ 冲刺回顾会议: (retrospective)

是由 **Scrum** 团队的**所有成员**参加。这次会议的焦点是对整个迭代进行回顾。细节包括: 什么进行顺利, 缺少什么, 需要改变什么等等。团队就未来的迭代改进计划达成一致。这个会议时间框为一个月的迭代, **3 个小时**, 比迭代评审时间短。

如上图所示, ,所有的会议都会在每次迭代中重复。

冲刺回顾是针对迭代末期进行的时间盒(有指定时间限制)会议, 目的是认识团队可以如何提高他们的工作方式, 就未来的迭代改进计划达成一致, 该会议:

- 针对冲刺末期召开;
- 被时间盒定义到三~四个小时按月冲刺和较短的时间段;
- 由包括开发团队, 产品负责人, **ScrumMaster**, 和企业的利益相关者的整个团队出席;
- 在冲刺回顾中, 团队将认识到他们做的好的领域以及有待改进的领域。
- 来自于回顾会议的反馈对实施持续改进策略和最大化团队交付价值非常关键。
- 细节包括: 什么进行顺利, 缺少什么, 需要改变什么等等……

## ⑤ 待办事项梳理(Grooming)

**Scrum** 团队在冲刺中经常会面进行待办事项的梳理。

梳理或细分是一种逐步完善待办事项的方法, 所以它会保留现有信息同时反映利益相关者的需要。该会议有助于:

- 增加新用户故事;

- 丢弃不相关的用户故事；
- 估算新增加的用户故事；
- 重新估算用户故事；
- 对用户故事进行优先级重排序；
- 史诗分解成更小的用户故事。

需要记住的点：

- 梳理会议提供了调整估算范围的最佳时机；
- 利益相关者的期望通过对产品待办事项进行与时俱进的更新来管理；
- 已经完成优先级排序和更新的产品待办事项应该作为冲刺评审会议的一部分由利益相关者来评审；
- 来自于运营和维护问题的反馈需要被考虑，新需求必须添加到产品待办事项中；
- 识别出的现有缺陷经过分析后，需要确保他们在梳理会议上被讨论。

### 三、其他常考知识点

#### 1、敏捷教练

它是指掌握了敏捷知识和经验的人员其在组织和团队转型中能够发挥培训、辅导和指导的作用。

敏捷教练可以是内部教练或外部教练，教练需要：

- 1) 跟不同团队共事时具备平衡视角；每个团队具备不同的进展节奏，可能会面临制约，需要帮助去克服；
- 2) 忠于团队成员价值；
- 3) 认识社会心理及团队复杂性；
- 4) 运用有效方法解决团队面临的问题；
- 5) 开发方法进行非侵入型干预从而改变团队动力；
- 6) 学习真正需要什么才能让人们作为一个团队去工作。

#### 2、仆人式领导

仆人式领导是一种为团队赋权的方法。仆人式领导是通过为团队服务来领导团队的实践，它注重理解和关注团队成员的需要和发展，旨在使团队尽可能达到最高绩效。仆人式领导的作用是促进团队发现和定义敏捷。仆人式领导实践并传播敏捷。仆人式领导按照以下顺序从事项目工作：

##### 1) 目的

与团队一起定义“为什么”或目的，以便他们能围绕项目目标进行合作互动。整个团队在项目层面而不是在人员层面优化。

##### 2) 人员

目标确立后，鼓励团队创造一个人人都能成功的环境。要求每个团队成员在项目工作中做出贡献。

##### 3) 过程

不要计划遵循“完美”的敏捷过程，而是要注重结果。如果跨职能团队能够常常交付完成的价值并反思产品和过程，团队就是敏捷的。团队将其过程称作什么并不重要。

##### 4) 特征



以下仆人式领导的特征让项目领导变得更加敏捷，促进团队的成功：

- A. 提升自我意识；
- B. 倾听；
- C. 为团队服务；
- D. 帮助他人成长；
- E. 引导与控制；
- F. 促进安全、尊重与信任；
- G. 促进他人精力和才智提升

### 3、优先级技术-MoSCoW

MoSCoW 技术是进行需求优先级排序的敏捷方法。在这种技术下，需求基于以下方面排序：

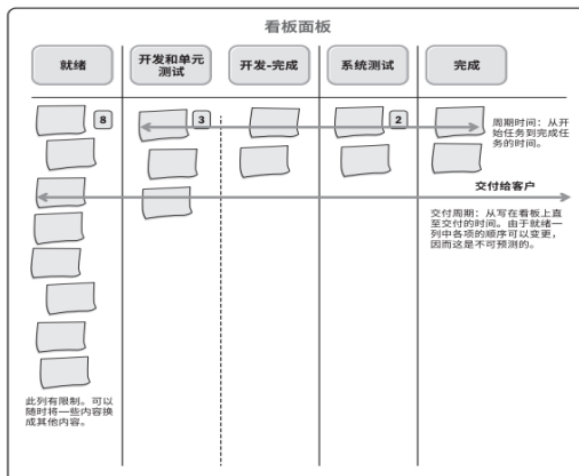
- 1) **Must** 必须有-这些需求是强制性的
- 2) **Should** 应该有-这些需求不是强制性的，但是高度渴望的
- 3) **Could** 可以有-这些需求如果满足会很好
- 4) **Won't** 不会有-当下可以不去满足，但是将来可以加入

在开始新一轮时间箱前，会有一个新的 **MUSTs** 加入。这些可能是新的需求，或者现有需求被调整优先级进而转移成为 **MUSTs**。

### 4、Kanban 看板

看板在项目实施期间作为信息发射源运用，它有助于相关干系人去了解冲刺或迭代的当下状态。

- 1) 看板是一个跟精益和及时制生产相关的概念。
    - 任务板被细分成段来反映关键活动。
    - 故事是由索引卡或代表的便利贴来表示。
    - 卡的状态由它在任务板上的位置来表示，并随着项目进展从开始到结束变化。
    - 看板帮助团队意识到他们是如何工作以及下一步要做什么。让团队形成自我指挥。
  - 2) **kanban** 卡片
    - **Kanban** 任务板上的每一张卡片就是 **kanban** 卡片。
    - **Kanban** 卡片用来显示迭代过程。
    - **Kanban** 任务板上的卡片呈现在开发周期的不同环节中移动的工作部件。
    - **Kanban** 卡片反映所有需要被跟踪的事物。例如：用户故事，缺陷，任务。
    - 在用户故事定义完整前，相关干系人需要对用户故事必须经历的部分进行评估。
  - 3) 简化的看板面板简化的看板面板有 3 列：
    - 待完成
    - 进展中
    - 已完成
- 任务用卡片表示，卡片状态展示在其中一列的下方。



## 5、五问法 (五个为什么 5whys)

五问法是一种通过不断重复询问为什么来识别问题根本原因的技术。每个为什么的答案成为了识别下一个为什么的驱动力。

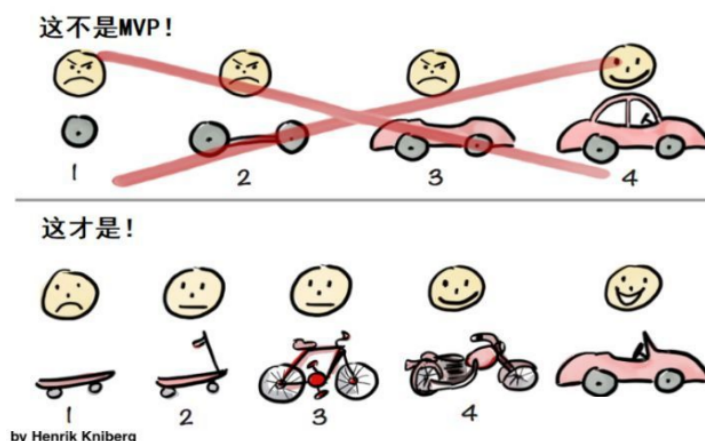
确切地说，不是强制的使用五个“为什么”来深入到问题的根源，“五”只是一个指示性的数字。这种技术同鱼骨图结合起来使用提供了一个问题解决可视化过程。

## 6、DoD 完成的定义

它是团队需要满足的所有标准的核对单，只有可交付成果满足该核对单才能视为准备就绪可供客户使用。

## 7、MVP

- 最小可交付价值
- 最小可售单元
- 最小可行性产品



8、WIP（在制品）

在敏捷开发中，WIP 限制决定了每种情况下的 workflows 中可以存续的最大工作量。限制进行中的工作数量可以更容易辨识团队 workflows 中的无效工作。在情况变得更糟前，一个团队在持续交付通道中的瓶颈是很容易辨别的。

WIP 限制通过强制让团队聚焦在更小的一套任务中来改善吞吐量和减少“将要完成”的工作量。从根本上来讲，WIP 限制鼓励的是“完成”的文化。更重要的是，WIP 限制可以让阻碍和瓶颈显而易见。当有明确指示现有工作遇到瓶颈时，团队可以聚焦在阻塞问题上尽快的理解、实施和解决。一旦消除阻塞，团队中的工作将再次开始流动。这些优势可以确保在最短的时间内向用户交付有价值的增量，从而使得 WIP 限制成为敏捷开发中一个非常有价值的工具。

9、用户故事

As a    [ User Role ],  
I want    [ Activities ],  
So that    [ Reason/Value ].

作为        [ 用户角色 ],  
我想要        [ 活动 ],  
以便        [ 原因 / 价值 ].

