

Scrum开发流程指导

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 拟制人 | 陈曦 |  | 创建日期 | 2016-10-28 |
| 当前版本 | V1.0 |  | 更新日期 | 2016-10-28 |

浙江力太科技有限公司

版权所有 侵权必究

**变更**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 作者 | 版本 | 变更参考 |
| 2016-10-28 | 陈曦 | 1.0 | 原始文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**审核**

|  |  |
| --- | --- |
| 姓名 | 职位 |
|  |  |
|  |  |
|  |  |
|  |  |

**分发**

|  |  |  |
| --- | --- | --- |
| 拷贝编号 | 姓名 | 位置/岗位 |
|  |  |  |
|  |  |  |
|  |  |  |

目录

[1 Scrum简介 5](#_Toc465937518)

[1.1 起源 5](#_Toc465937519)

[2 Scrum的角色 6](#_Toc465937520)

[2.1 产品负责人（Product Owner） 6](#_Toc465937521)

[2.2 团队主持人（Scrum Master） 7](#_Toc465937522)

[2.2.1 Scrum Master 服务于产品负责人 7](#_Toc465937523)

[2.2.2 Scrum Master 服务于开发团队 7](#_Toc465937524)

[2.3 团队成员（Scrum Team） 8](#_Toc465937525)

[3 Scrum的工件 8](#_Toc465937526)

[3.1 用户故事（User Story） 8](#_Toc465937527)

[3.1.1 Ron Jeffries的3个C 9](#_Toc465937528)

[3.1.2 用户故事的六个特性- INVEST 9](#_Toc465937529)

[3.1.3 用户故事完成的标准 10](#_Toc465937530)

[3.1.4 拆分用户故事 10](#_Toc465937531)

[3.2 产品待办事项（Product Backlog） 10](#_Toc465937532)

[3.3 Sprint待办事项（Sprint Backlog） 12](#_Toc465937533)

[3.4 燃尽图（Burn-down Chart) 12](#_Toc465937534)

[3.5 Sprint燃尽图（Sprint Burn-down Chart) 12](#_Toc465937535)

[3.6 发布燃尽图（Release Burn-down Chart） 13](#_Toc465937536)

[4 Scrum的活动 13](#_Toc465937537)

[4.1 产品梳理（Product Backlog Refinement） 13](#_Toc465937538)

[4.1.1 Sprint 14](#_Toc465937539)

[4.2 计划会议（Sprint Planning Meeting） 14](#_Toc465937540)

[4.2.1 敏捷估算 16](#_Toc465937541)

[4.2.2 任务分解 17](#_Toc465937542)

[4.3 每日立会（Daily Scrum Meeting） 18](#_Toc465937543)

[4.3.1 每日站会的目的 18](#_Toc465937544)

[4.3.2 每日站会的时间和地点 18](#_Toc465937545)

[4.3.3 每日站会的纪律 18](#_Toc465937546)

[4.3.4 任务看板 20](#_Toc465937547)

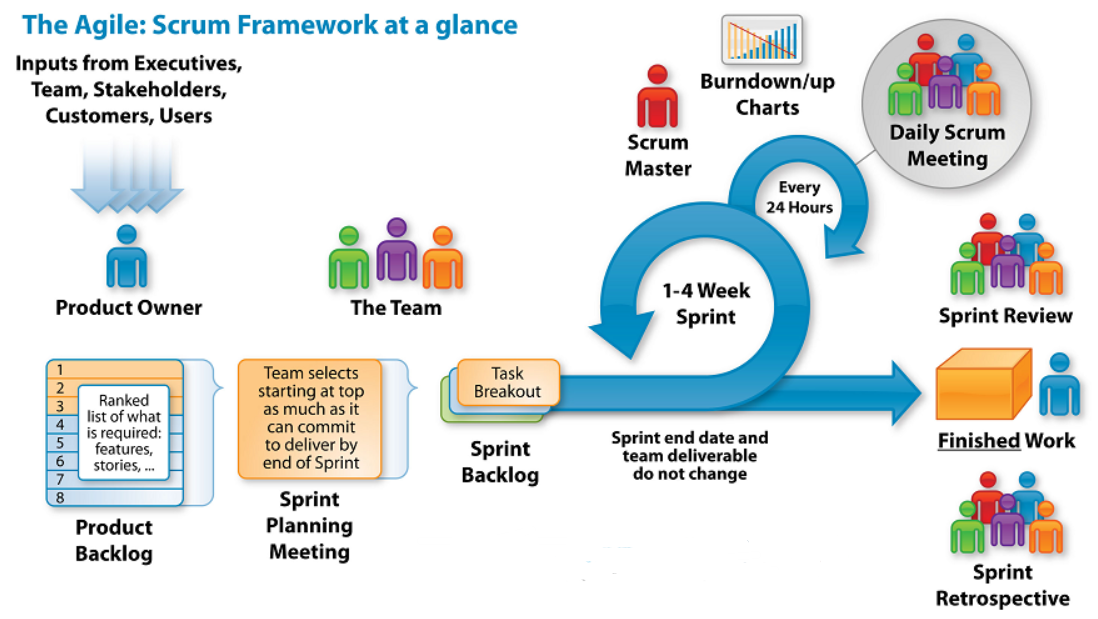
[4.4 评审会议（Sprint Review Meeting） 21](#_Toc465937548)

[4.5 回顾会议（Sprint Retrospective Meeting） 22](#_Toc465937549)

# Scrum简介

Scrum 是一个用于开发和维持复杂产品的框架 ，是一个增量的、迭代的开发过程。在这个框架中，整个开发过程由若干个短的迭代周期组成，一个短的迭代周期称为一个Sprint，每个Sprint的建议长度是2到4周(互联网产品研发可以使用1周的Sprint)。在Scrum中，使用产品Backlog来管理产品的需求，产品backlog是一个按照商业价值排序的需求列表，列表条目的体现形式通常为用户故事。Scrum团队总是先开发对客户具有较高价值的需求。在Sprint中，Scrum团队从产品Backlog中挑选最高优先级的需求进行开发。挑选的需求在Sprint计划会议上经过讨论、分析和估算得到相应的任务列表，我们称它为Sprint backlog。在每个迭代结束时，Scrum团队将递交潜在可交付的产品增量。 Scrum起源于软件开发项目，但它适用于任何复杂的或是创新性的项目。

Scrum流程如下图：



1. Scrum流程图

## 起源

Scrum原始含义是指英式橄榄球次要犯规时在犯规地点对阵争球。争球双方各有8个队员参与，各方出3名前锋队员，并肩各站成一横排，面对面躬身互相顶肩，中间形成一条通道，其他前锋队员分别站在后面，后排队员用肩顶住前锋队员的臀部，组成3、2、3或3、4、1阵形。然后，由犯规队的对方队员在对阵一侧1码外，用双手低手将球抛入通道，不得有利于本队。当球抛入通道时，前排的3对前锋队员互相抗挤，争相踢球给本方前卫或后卫队员，前卫和后卫队员必须等候前锋将球踢回后，方可移动。

[](file:///C:\Users\chenxi\Desktop\Scrum,敏捷开发,敏捷实践集%20-%20Scrum中文网_files\ScrumCN_Scrum.jpg)

1986年，竹内弘高和 野中郁次郎在New New Product Development Game文章首次提到将Scrum应用与产品开发，他们指出：

传统的”接力式”的开发模式已经不能满足快速灵活的市场需求，而整体或”橄榄球式”的方法——团队作为一个整体前进，在团队的内部传球并保持前进，这也许可以更好的满足当前激烈的市场竞争。

敏捷思想深受日本工业界最佳实践的影响，尤其是丰田和本田公司推行的精益原则，以及竹内弘高和 野中郁次郎开发的知识管理策略。受到以上思想的影响，以及对世界范围内软件项目的研究，Jeff Sutherland在 1993年首次在Easel公司定义了用于了软件开发行业的Scrum流程，并开始实施。

1995年Jeff Sutherland和Ken Schwaber规范化了Scrum框架，并在OOPSLA 95上公开发布。

2001年 敏捷宣言及原则发布、敏捷联盟成立，Scrum是其中一种敏捷方法。

2001年，Ken Schwaber和Mike Beedle推出第一本Scrum书籍《Scrum敏捷软件开发》。

2002年Ken Schwaber 和Mike Cohn共同创办了Scrum联盟。

Scrum框架

# Scrum的角色

## 产品负责人（Product Owner）

产品负责人负责最大化产品以及开发团队工作的价值。实现这一点的方式会随着组织、Scrum团队以及单个团队成员的不同而不同。

产品负责人是管理产品待办事项列表的唯一责任人。产品待办事项列表的管理包括:

* 清晰地表达产品代办事项列表条目
* 对产品代办事项列表中的条目进行排序，最好地实现目标和使命
* 确保开发团队所执行工作的价值
* 确保产品代办事项列表对所有人可见、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代办事项列表中的条目达到一定程度的理解

产品负责人可以亲自完成上述工作，也可以让开发团队来完成。然而，产品负责人是负责任者。

产品负责人是一个人，而不是一个委员会。产品负责人可能会在产品代办事项列表中体现一个委员会的需求，但要想改变某条目的优先级必须先说服产品负责人。

为保证产品负责人的工作取得成功，组织中的所有人员都必须尊重他的决定。产品负责人所作的决定在产品待办事项列表的内容和排序中要清晰可见。任何人都不得要求开发团队按照另一套需求开展工作，开发团队也不允许听从任何其他人的指令。

## 团队主持人（Scrum Master）

Scrum Master 负责确保 Scrum 被理解并实施。为了达到这个目的，Scrum Master要确保 Scrum 团队遵循 Scrum 的理论、实践和规则。Scrum Master是Scrum团队中的服务式领导。

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。 Scrum Master 通过改变这些交互来最大化 Scrum 团队所创造的价值。

### Scrum Master 服务于产品负责人

Scrum Master 以各种方式服务于产品负责人，包括:

* 找到有效管理产品代办事项列表的技巧
* 清晰地和开发团队沟通愿景、目标和产品代表事项列表条目
* 教导开发团队创建清晰简明的产品代表事项列表条目
* 在经验主义环境中理解长期的产品规划
* 理解并实践敏捷
* 按需推动Scrum活动

### Scrum Master 服务于开发团队

Scrum Master 以各种方式服务于开发团队，包括:

* 指导开发团队自组织和跨职能
* 教导并领导开发团队创造高价值的产品
* 移除开发团队进展过程中的障碍
* 按需推动Scrum活动
* 在 Scrum 还未完全被采纳和理解的组织环境下指导开发团队

Scrum Master 以各种方式服务于组织，包括:

* 领导并指导组织采用 Scrum
* 在组织范围内计划 Scrum 的实施
* 帮助员工及干系人理解并实施 Scrum 和经验性产品开发
* 发起能提升Scrum 团队生产力的变革
* 与其他 Scrum Master 一起工作，帮助组织更有效的应用Scrum

## 团队成员（Scrum Team）

团队成员包含了专业人员，负责在每个 Sprint 的结尾交付潜在可发布的“完成”产品增量。只有开发团队的成员才能创造增量。

开发团队由组织构建并授权，来组织和管理他们的工作。所产生的协同工作能最大化开发团队的整体效率和效力。开发团队有以下几个特点：

* 他们是自组织的，没有人可以告诉开发团队如何把产品代办列表变成潜在可发布的功能。
* 开发团队是跨职能的，团队作为一个整体拥有创造产品增量所需要的全部技能。
* Scrum 不认可开发团队成员的头衔，无论承担哪种工作他们都是开发者。此规则无一例外。
* 开发团队中的每个成员可以有特长和专注领域，但是责任归属于整个开发团队
* 开发团队不包含如测试或业务分析等负责特定领域的子团队。

开发团队最佳规模是小到足以保持敏捷性，大到足以完成重要工作。少于 3 人的开发团队没有足够的交互，因而所获得的生产力增长也不会很大。小团队在 Sprint 中可能会 受到技能限制，从而导致无法交付可发布的产品增量。大于 9 人的团队需要过多的协调沟通工作。大型团队会产生太多复杂性，不便于经验过程管理。产品负责人和 Scrum Master 的角色不包含在此数字中，除非他们也参与执行 Sprint 代表事项列表中的工作。

# Scrum的工件

## 用户故事（User Story）

用户故事是从用户的角度来描述用户渴望得到的功能。一个好的用户故事包括三个要素：

1. 角色：谁要使用这个功能。
2. 活动：需要完成什么样的功能。
3. 商业价值：为什么需要这个功能，这个功能带来什么样的价值。

用户故事通常按照如下的格式来表达：

英文：As a ， I want to ， so that .

中文：作为一个<角色>， 我想要<活动>， 以便于<商业价值>

举例：作为一个“网站管理员”，我想要“统计每天有多少人访问了我的网站”，以便于“我的赞助商了解我的网站会给他们带来什么收益”。

需要注意的是用户故事不能够使用技术语言来描述，要使用用户可以理解的业务语言来描述。

### Ron Jeffries的3个C

关于用户故事，Ron Jeffries用3个C来描述它：

卡片（Card） – 用户故事一般写在小的记事卡片上。卡片上可能会写上故事的简短描述，工作量估算等。

交谈（Conversation）- 用户故事背后的细节来源于和客户或者产品负责人的交流沟通。

确认（Confirmation）- 通过验收测试确认用户故事被正确完成。

### 用户故事的六个特性- INVEST

一个好的用户故事应该遵循INVEST原则。

* 独立性（Independent）— 要尽可能的让一个用户故事独立于其他的用户故事。用户故事之间的依赖使得制定计划，确定优先级，工作量估算都变得很困难。通常我们可以通过组合用户故事和分解用户故事来减少依赖性。
* 可协商性（Negotiable）— 一个用户故事的内容要是可以协商的，用户故事不是合同。一个用户故事卡片上只是对用户故事的一个简短的描述，不包括太多的细节。具体的细节在沟通阶段产出。一个用户故事卡带有了太多的细节，实际上限制了和用户的沟通。
* 有价值（Valuable）— 每个故事必须对客户具有价值（无论是用户还是购买方）。一个让用户故事有价值的好方法是让客户来写下它们。一旦一个客户意识到这是一个用户故事并不是一个契约而且可以进行协商的时候，他们将非常乐意写下故事。
* 可以估算性（Estimable）—开发团队需要去估计一个用户故事以便确定优先级，工作量，安排计划。但是让开发者难以估计故事的问题来自：对于领域知识的缺乏（这种情况下需要更多的沟通），或者故事太大了（这时需要把故事切分成小些的）。
* 短小（Small）— 一个好的故事在工作量上要尽量短小，最好不要超过10个理想人/天的工作量，至少要确保的是在一个迭代或Sprint中能够完成。用户故事越大，在安排计划，工作量估算等方面的风险就会越大。
* 可测试性（Testable）—一个用户故事要是可以测试的，以便于确认它是可以完成的。如果一个用户故事不能够测试，那么你就无法知道它什么时候可以完成。一个不可测试的用户故事例子：软件应该是易于使用的。

### 用户故事完成的标准

团队每位成员及团队相关人员应对每个用户故事的”完成”标准有一致的定义，产品backlog包中每个用户故事都要描述完成标准（或者称为”如何演示”），Sprint计划会议上大家首先要讨论用户故事的完成标准并达成一致。

### 拆分用户故事

如果用户故事过大，将不能在一个Sprint内完成；有时用户故事虽然不大，但是对于正在规划的Sprint中剩下的时间不够完成这个用户故事，这些情况都需要拆分用户故事。另外，对大型用户故事进行拆分，也有助于更准确的估计。常见的用户拆分用户故事的指导原则有：

1. 按照用户角色拆分
2. 按照业务操作边界拆分
3. 按照数据边界拆分
4. 按照公共功能边界拆分
5. 按照验收条件拆分
6. 拆分具有混合优先级的用户故事

拆分用户故事时需要注意的是：

* 不要把用户故事拆分成任务。例如将故事拆分为编写Web界面、完成后端核心处理，或将故事拆分为设计、编码和测试，这样的拆分结果对用户或测试人员来说毫无价值。
* 不要从系统设计和模块的角度拆分用户故事。例如将系统拆分为用户界面、状态机、定时器、存储等用户故事，这些用户故事中任何一个都不构成一个系统功能。

## 产品待办事项（Product Backlog）

产品待办事项列表是一个排序的列表，包含所有产品需要的东西，也是产品需求变动的唯一来源。产品负责人负责产品待办事项列表的内容、可用性和优先级。

产品待办事项列表是一个持续完善的清单， 最初的版本只列出最初始的和众所周知的需求。 产品待办事项列表根据产品和开发环境的变化而演进。待办事项列表是动态的，它经常发生变化以识别使产品合理、有竞争力和有用所必需的东西。只要产品存在，产品待办事项列表就存在。

产品待办事项列表列出了所有的特性、功能、需求、改进方法和缺陷修复等对未来发布产品进行的改变。产品待办事项列表条目包含描述、次序和估算的特征。

产品待办事项列表通常以价值、风险、优先级和必须性排序。它是一个按照优先级由高到低排列的一个序列，每个条目有唯一的顺序。排在顶部的产品待办事项列表条目需要立即进行开发。排序越高，产品待办事项列表条目越紧急，就越需要仔细斟酌，并且对其价值的意见越一致。

排序越高的产品待办事项列表条目比排序低的更清晰、更具体。根据更清晰的内容和 更详尽的信息就能做出更准确的估算。优先级越低，细节信息越少。开发团队在接下来的 Sprint 中将要进行开发的产品待办事项列表条目是细粒度的，已经被分解过，因此，任何 一个条目在 Sprint 的时间盒内都可以被”完成”。开发团队在一个 Sprint 中可以”完 成”的产品待办事项列表条目被认为是”准备好的”或者”可执行的”，能在 Sprint 计 划会议中被选择。

随着产品的使用、价值的获取以及市场的反馈，产品待办事项列表变成了更大、更详 尽的列表。因为需求永远不会停止改变，所以产品待办事项列表是个不断更新的工件。业 务需求、市场形势和技术的变化都会引起产品待办事项列表的变化。

若干个 Scrum 团队常常会一起开发某个产品。但描述下一步产品开发工作的产品待办事项列表只能有一个。那么这就需要使用对产品待办事项列表条目进行分组的属性。

通过产品Backlog地梳理来增添细节、估算和排序。这是一个持续不断 的过程，产品负责人和开发团队协作讨论产品代表事项列表条目的细节。在产品待办事项列表梳理的时候，条目会被评审和修改。然而， 产品负责人可以随时更新产品代办事项列表条目或酌情决定。

梳理在 Sprint 中是一项兼职活动，在产品负责人和开发团队之间展开。通常，开发 团队有自行优化的领域知识。然而，何时如何完成优化是 Scrum 团队的决定。优化通常占用不超过开发团队 10%的时间。

开发团队负责所有的估算工作。产品负责人可以通过协助团队权衡取舍来影响他们的 决定。但是，最后的估算是由执行工作的人来决定的。



1. 产品待办事项

## Sprint待办事项（Sprint Backlog）

Sprint 代办事项列表是一组为当前 Sprint 选出的产品代办事项列表条目，外加交付 产品增量和实现 Sprint 目标的计划。Sprint 代办事项列表是开发团队对于哪些功能要包 含在下个增量中，以及交付那些功能所需工作的预计。

Sprint 代办事项列表定义了开发团队把产品代办事项列表条目转换成”完成”的增量 所需要执行的工作。Sprint 代办事项列表使开发团队确定的、达到 Sprint 目标所需的工 作清晰可见。

Sprint 代办事项列表是一份足够具体的计划，使得进度上的改变能在每日例会中得到 理解。开发团队在整个 Sprint 中都会修改 Sprint 代办事项列表，Sprint 代办事项列表也 会在 Sprint 的进程中慢慢显现，比如开发团队按照计划工作并对完成 Sprint 目标所需的 工作有更多的了解。

当出现新工作时，开发团队需要将其追加到 Sprint 待办事项列表中去。随着任务进 行或者被完成，需要更新每项任务的估算剩余工作量。如果计划中某个部分失去开发的意 义，就可以将其除去。在 Sprint 内只有开发团队可以对 Sprint 待办事项列表进行修改。 Sprint 待办事项列表是高度可见的，是对团队计划在当前 Sprint 内完成工作的实时反 映，并且，该列表只属于开发团队。

Product Backlog 功能点被放到Sprint的固定周期中，Sprint Backlog 会因为如下原因发生变化:

1. 随着时间的变化，开发团队对于需求有了更好的理解，有可能发现需要增加一些新的任务到Sprint Backlog中。

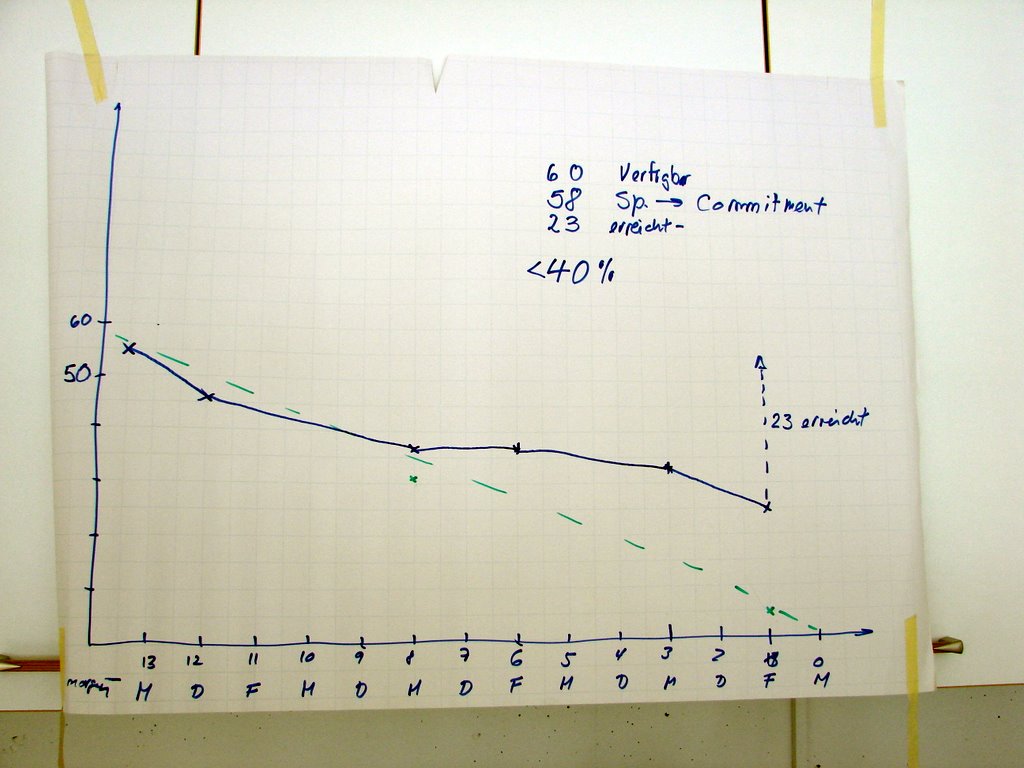
2. 程序缺陷做为新的任务加进来，这个都做为承诺提交任务中未完成的工作。

Product Owner也许会和Scrum team一起工作，以帮助team更好的理解Sprint的目标，Scrum Master和team也许会觉得小的调整不会影响sprint的进度，但会给客户带来更多商业价值。

## 燃尽图（Burn-down Chart)

## Sprint燃尽图（Sprint Burn-down Chart)

Sprint Burndown Chart 显示了Sprint中累积剩余的工作量，它是一个反映工作量完成状况的趋势图。 图中Y轴代表的是剩余工作量，X轴代表的是Sprint的工作日。



1. Sprint燃尽图

在Sprint开始的时候，Scrum Team会标示和估计在这个Sprint需要完成的详细的任务。所有这个Sprint中需要完成，但没有完成的任务的工作量是累积工作量，团队会根据进展情况每天更新累积工作量，如果在Sprint结束时，累积工作量降低到0，Sprint就成功结束。

由于在Sprint的刚开始的时候，增加的任务工作量可能大于完成的任务工作量，所以燃尽图有可能略微呈上升趋势。

## 发布燃尽图（Release Burn-down Chart）

在Scrum项目中，团队通过每个Sprint结束时更新的发布燃尽图来跟踪整个发布计划的进展。发布燃尽图记录了在一段时间内产品Backlog的总剩余估算工作量的变化趋势。X轴代表的项目周期，以Sprint为单位， Y轴代表的是剩余工作量，通常以用户故事点、理想人天或者team-days为单位。

# Scrum的活动

## 产品梳理（Product Backlog Refinement）

产品待办事项通常会很大，也很宽泛，而且想法会变来变去、优先级也会变化，所以产品待办事项列表梳理是一个贯穿整个Scrum项目始终的活动。该活动包含但不限于以下的内容：

* 保持产品待办事项列表有序
* 把看起来不再重要的事项移除或者降级
* 增加或提升涌现出来的或变得更重要的事项
* 将事项分解成更小的事项
* 将事项归并为更大的事项
* 对事项进行估算

产品待办事项列表梳理的一个最大好处是为即将到来的几个Sprint做准备。为此，梳理时会特别关注那些即将被实现的事项。需要考虑不少因素，这包括但不限于以下的内容：

理想情况下，下一个Sprint的备选事项都应该提升”商业价值”。 开发团队需要能够在一个Sprint内完成每一个事项。每个人都需要清楚预期产出是什么。

产品开发决定了，有可能需要其它的技能和输入。因此，产品待办事项列表梳理最好是所有团队成员都参与的活动，而不单单是产品负责人。

### Sprint

Scrum是一种迭代和增量式的产品开发方法，Scrum通过Sprint来实现迭代。一个Sprint是指一个1周－4周的迭代，它是一个时间盒。Sprint的长度一旦确定，保持不变。Sprint的产出是”完成”的、可用 的、潜在可发布的产品增量。Sprint 在整个开发过程中的周期一致。新的 Sprint 在上一 个 Sprint 完成之后立即开始。  Sprint 包含并由 Sprint 计划会议、每日站会、开发工作、Sprint 评审会议和 Sprint 回顾会议构成。

Scrum采用迭代增量的方式，是因为需求是涌现的，我们对产品和需求的理解是渐进式的，Sprint长度越长，我们需要预测的越多，复杂度会提升、风险也会增加，所以Sprint的长度最多不超过4周。越来越多的团队使用2周的Sprint，很多市场变化快、竞争激烈的领域，比如互联网和移动互联网产品开发团队也会使用1周的迭代。

在Sprint进行过程中，如下内容不能发生变化：

* Sprint的目标
* Sprint的质量目标和验收标准
* 开发团队的组成

## 计划会议（Sprint Planning Meeting）

每个Sprint都以Sprint计划会议作为开始，这是一个固定时长的会议，在这个会议中，Scrum团队共同选择和理解在即将到来的Sprint中要完成的工作。

整个团队都要参加Sprint计划会议。针对排好序的产品待办事项列表(Product Backlog)，产品负责人和开发团队成员讨论每个事项，并对该事项达成共识，包括根据当前的”完成的定义”，为了完成该事项所需要完成的所有事情。所有的Scrum会议都是限定时⻓的。Sprint计划会议推荐时⻓是Sprint中的每周对应两⼩时或者更少(译者注:比如，一个Sprint包含2个星 期，则Sprint计划会议时长应为4个小时或者更少)。因为会议是限制时⻓长的，Sprint计划会议的成功⼗分依赖于产品待办事项列表的质量。这就是产品待办事项列表梳理十分重要的原因。

在Scrum中，Sprint计划会议有两部分:

1.决定在Sprint中需要完成哪些工作

2.决定这些工作如何完成

第一部分:需要完成哪些工作?

在会议的第一部分，产品负责人向开发团队介绍排好序的产品待办事项，整个Scrum团队共同理解这些工作。

Sprint中需要完成的产品待办事项数目完全由开发团队决定。为了决定做多少，开发团队需要考虑当前产品增量的状态，团队过去的工作情况，团队当前的生产能力，以及排好序的产品待办事项列表。做多少工作只能由开发团队决定。产品负责人或任何其它人，都不能给开发 团队强加更多的工作量。

通常Sprint都有个目标，称作Sprint目标。这将十分有效地帮助大家更加专注于需要完成的工 作的本质，而不必花太多精力去关注那些对于我们需要完成的工作并不重要的⼩小细节。

第二部分:如何完成工作?

在会议的第二部分⾥里，开发团队需要根据当前的”完成的定义”一起决定如何实现下一个产品增 量。他们进⾏行⾜足够的设计和计划，从而有信心可以在Sprint中完成所有工作。头几天的工作会 被分解成⼩小的单元，每个工作单元不超过一天。之后要完成的工作可以稍⼤大些，以后再对它 们进⾏行分解。

决定如何完成工作是开发团队的职责，决定做什么则是产品负责人的职责。

在计划会议的第二部分，产品负责人可以继续留下来回答问题，以及澄清一些误解。不管怎样，团队应该很容易找到产品负责人。

Sprint计划会议的产出 Sprint计划会议最终需要Scrum团队对Sprint需要完成工作的数量和复杂度达成共识，并预期在一个合理的条件范围内完成它们。开发团队预测并共同承诺他们要完成的工作量。 总而⾔言之:在Sprint计划会议中，开发团队和产品负责人一起考虑并讨论产品待办事项，确保他们对这些事项的理解，选择一些他们预测能完成的事项，创建足够详细的计划来确保他们能够完成这些事项。

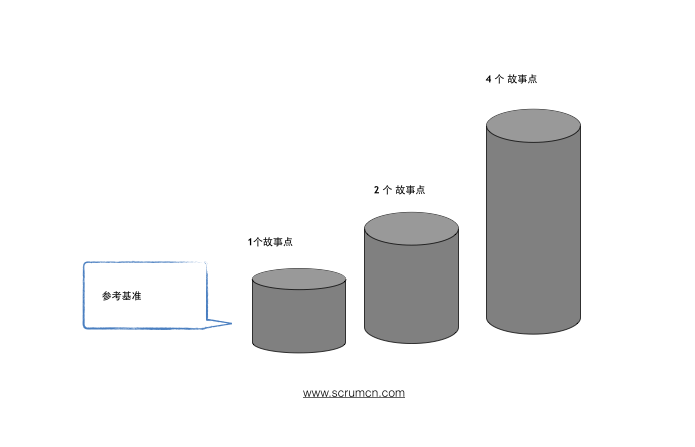
最终产生的待办事项列表就是”Sprint待办事项列表(Sprint Backlog)”。

### 敏捷估算

#### 故事点（Story Point）

同一项工作，不同能力的人完成它花费的时间显然是不一样的。如果我们要估算从家到公司的绝对距离时多少公里，您可能不一定知道，但是如果您时坐地铁上班，从家里到公司有多少站，你一定很容易知道，当我们知道有多少站之后，我们就可以大概清楚路上需要花多长时间了。

敏捷估算时，我们不会估算绝对时间和周期，我们估算大小，和相对值，也就是倍数。敏捷估算时，我们使用故事点作为计量单位，它是一个倍数，我们会先找一个我们认为最小的一个功能的大小作为参考基准，定义为1个故事点，把其它的故事和它做比较，如果是2倍大小，就是2个故事点，如果是5倍大小，就是5个故事点。

[](http://www.scrumcn.com/agile/wp-content/uploads/2014/04/storypoint.png)

1. 故事点

#### 纸牌游戏

在Scrum的开发过程中，团队共担责任，集体承诺每个Sprint的工作，因此对于工作量的估算敏捷团队采用集体估算的方式。集体估算，通常采用估算扑克作为工具，团队通过玩估算游戏进行集体估算。使用估算扑克来做工作量估算是最有效，也是非常有趣的一种估算方式。估算扑克由一组类似斐波纳契数列的数字组成，这些数字包括：0，0.5，1，2，3，5，8，20，40，?，∞， 每幅扑克有四组这样的数字，可供4个人使用。

估算扑克的使用方法：

1. 每个团队成员拿到一组卡片，包括0，0.5，1，2，3，5，8，13，20，40，?，∞，共计12张。
2. 产品负责人或者一名团队成员扮演阅读者的角色，他负责阅读需要估算产品Backlog的条目，并且询问大家是否有疑问。
3. 团队讨论这个条目。
4. 当团队理解了这个条目之后，每个团队成员按照自己的想法给出估算结果，并且选择对应的扑克出牌，估算结果不能告诉其他人，出牌时数字朝下扣在桌面上。
5. 所有人都出牌之后，阅读者向大家确认是否都已经确定估算结果，确认后，数”1，2，3″，大家同时展示估算结果。
6. 团队评估不同的估算结果.我们是否想法一致？我们是否存在分歧？有没有什么是我没有考虑到的？讨论之后可以再估算一轮，最终团队需要达成一致。
7. 回到第二步，开始估算下一个条目。

#### 确定Sprint待办事项（Sprint Backlog）

要确定完成所有用户故事需要几个Sprint，就必须确定每个Sprint能完成几个故事点，即迭代开发速度。通过稳定的迭代开发速度，可以估算出迭代次数和项目长度。

**开发速度 = 一个Sprint完成的故事点数**

确定迭代开发速度的常用方法有：

1. 使用历史值。借鉴以往类似项目的开发速度。
2. 进行一次或几次迭代。比较理性的预测开发速度的方式是项目先进行1~3次迭代，通过参考前面1~3次迭代的开发速度来估算后继迭代周期的开发速度。
3. 进行测算。先对本次迭代的用户故事进行任务分解，估计出用户故事需要的总时间，然后反推本次迭代可以完成几个故事点。

确定了迭代开发速度后，即可从产品Backlog中根据优先级由高到低取出相应量的用户故事，形成本Sprint的Backlog包。

### 任务分解

一旦确定本Sprint内计划完成的用户故事，接着就需要把每个用户故事分解为可操作的任务。例如可以把任务分解为：需求、设计、实现和验证。

对每项任务需要花费的工作量进行估时，一般以人时为单位。

## 每日立会（Daily Scrum Meeting）

### 每日站会的目的

敏捷宣言强调个体交互重于过程和工具，敏捷原则阐述了面对面的沟通和自组织的团队这些敏捷的核心思想。Scrum的团队是一个自组织的团队，团队每天进行每日站会是团队面对面沟通和团队自组织的体现。Scrum的理论基础是通过保持过程透明性让参与过程的所有人了解真实状况，然后进行检查和调整，每日站会是Scrum过程进行每天的检查和调整的环节。

### 每日站会的时间和地点

Scrum定义了开展每日站会的一些基本的规则。每日站会必须每天在同一时间、同一地点召开，最好的方式是在团队的可视化的任务板前面召开。 任务板上可以看到当前Sprint的燃尽图（Burn Down Chart）和Sprint中每个任务的状态。

在每日站会开始之前，团队需要在任务板上更新任务的状态。这样的好处是在开会的时候，每个人都可以看到当前的进展情况。

每日站会是Scrum团队每天的第一件事情，这样可以让每个人在每天一开始就清楚的了解他一天的安排。对于跨国界的团队，存在时间差的情况，可以根据实际情况做调整。

### 每日站会的纪律

会议时间最多不超过15分钟。所有的团队成员自觉按时到场，因为会议很短，按时召开按时结束是很重要的。团队需要建立他们的工作协议来确保团队成员按时出席，并且遵守站会纪律，比如团队可以商量对于迟到的人员要有一些让他们改进的措施，比如适当的给一些罚金，多少由团队共同决定，这些钱如何支配也由团队共同决定， 或者做俯卧撑、挂一个迟到的牌子等等。

每日站会一定要站着开，每个人要精神集中，不能有懒散的表现。

每个人回答三个问题：

* 我昨天完成了什么任务？
* 我今天打算做什么任务？
* 我遇到了哪些障碍或困难？

同一时间只能有一个人发言，会上只说和这三个问题相关的话题，任何跑题的讨论，需要被ScrumMaster制止。一些的确需要讨论的问题，可以先记录下来，会后作为专题来讨论。

每日站会和传统的项目会议有如下几点不同：

1.不会有ScrumMaster或者其他任何人来指派任务。

2.团队成员不是向ScrumMaster汇报情况，每日站会是团队自己的会。

3.团队成员不会在会上讨论或者解决问题，大家会把问题记录下来，会后找相关的人讨论或召开具体的讨论会议。

4.任何团队之外的人不得发言或干扰会议。

Scrum的最基本原则是”Inspect and Adapt”(检视然后适应)，如果什么事情做得很好，问问自己为什么，然后寻找提升的办法。

如果每日站会没有效果，检查一下这些规则：你是不是每天在认真开每日站会？如果不是为什么？如果你改变了Scrum的一些基本的规则，你可能会面临一些风险，因为这些规则都是经过锤炼和项目考验的一些通用规则。所以第一步，你可以先按照书本上的方式来做。

一个好的每日站会有如下几个特点：

1.ScrumMaster不会逐个的问每个人问题，如果是，那么这个会议已经沦为了报告会。

2.团队成员互相交流，不是向ScrumMaster报告。

3.每日站会都会在15分钟以内完成。如果你遵守了规则并按照正确的方式开会，你就不需要再担心超时了。

4.站会结束后，ScrumMaster知道哪些问题需要帮助团队成员解决。

一个自组织的团队有一个非常明显的每天的节奏：Daily Scrum之前非常安静，每日站会之后会有一段活跃的讨论，到中餐前的时候就慢慢安静下来了。午饭之后会有另外一个阶段的活跃讨论，当下班前慢慢的安静下来。这就是一个自组织团队的脉冲。如果你能够感受到这个节奏，则说明团队是很健康的，每日站会起到了很好的效果。

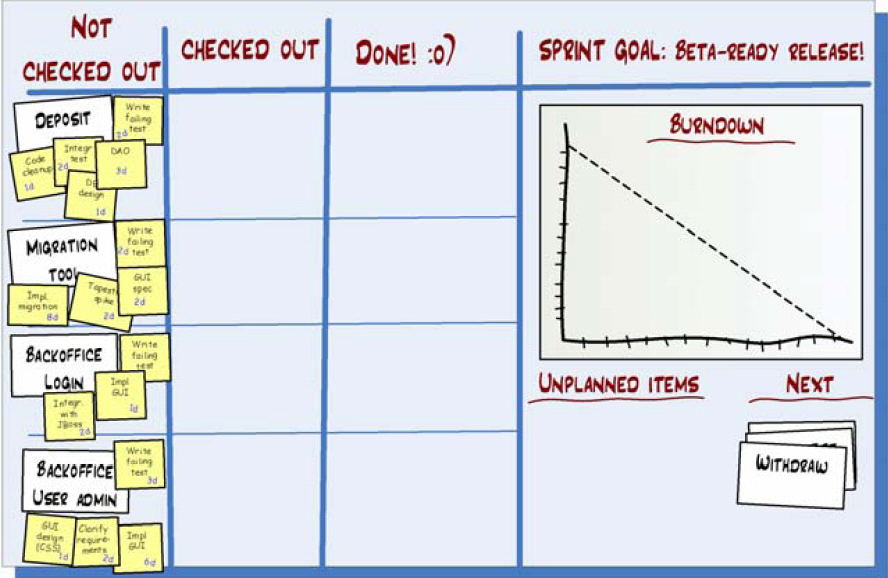


1. 每日站会

### 任务看板

一般任务看板分为三列：To Do还没做的， Doing正在做的， Done做完的。

有些团队的分工比较多，会出现一些中间状态，比如“还没做的/正在开发的/等待测试的/正在测试的/等待评审的”是一种典型的开发与测试分离的任务看板。



1. 任务看板

#### 任务看板的用法

任务看板可以帮助解决一些燃尽图解决不了的问题：

1. 有哪些故事正在做，还没有做，已经开工了但没完成……？任务看板的三列正好解决问题。

2. 最后剩下了哪些故事没完成？最左边剩下的就是。

3. 有没有人不是一个一个完成故事，而是同时开工了很多故事？（这个是最后很多故事都开工了但都差一点完成不了的主要原因）这个复杂一些，下面讲。

同时开工很多故事，很容易造成思绪散乱、最后全部完不成的情况，确切说瀑布模型就是同时开工所有故事的典范。

为了防止这一点，如果是手工做的任务看板，，那么注意中间一个”正在做的”列一点要窄一些，这样当里边的故事挤不下的时候，就是一个危险的很多故事都在开工的信号。

还有一种做法，是每个人只准放一个故事在中间，做完一个，才能再做一个。这个严格坚持有难度，因为经常遇到被卡住的情况，但是作为一种思路和精神应该尽量坚持。

4. 如果有跟进人，谁负责跟进哪个？可以在卡片上写上当前负责人、跟进人的名字。

通常的做法有很多种，比如每个人用自己颜色的即时贴，可以比较容易地看出每个人有多少个故事，分别处于什么状态。不过，这样就需要在”尚未开始的”里边提前分配人员了，不太利于后期的互动和互相关注；当然还可以在开始的时候重新用有颜色的即时贴重新抄一个，看大家的习惯了。

## 评审会议（Sprint Review Meeting）

Sprint结束时，Scrum团队和相关⼈人员一起评审Sprint的产出。所有Scrum会议都是限定时⻓长的，Sprint评审会议的推荐时⻓长是Sprint中的每一周对应一个小时(比如，一个Sprint 包含2个星期，则Sprint评审会议时⻓长为2个小时)。

讨论围绕着Sprint中完成的产品增量。由于Sprint的产出会涉及到一些⼈人的”利益”，因此一个明智的做法是邀请他们参加这个会议，这会很有帮助。这个会议是个⾮非正式的会议，帮助⼤大家了解我们目前进展到哪里，并一起讨论我们下一步如何推进。每个人都可以在Sprint评审会议上发表意见。当然，产品负责⼈人会对未来做出最终的决定，并适当地调整产品待办事项列表 (Product Backlog)。

团队会找到他们自己的方式来开Sprint评审会议。通常会演⽰示产品增量，整个小组也会经常讨论他们在Sprint中观察到了什么、有哪些新的产品想法出现。他们还会讨论产品待办事项列表 的状态、可能的完成日期以及在这些日期前能完成什么。

Sprint评审会议向每个⼈人展⽰示了当前产品增量的概况。因此，通常都会在Sprint评审会议中调 整产品待办事项列表。

Sprint评审会议应该注意以下几点：

* 尽量邀请项目相关人员都参加
* 集中精力演示可以实际工作的代码，让演示关注与业务层次，不要演示一大堆细碎的bug修复和微不足道的特性。
* 时间通常是30~60分钟，不要超过2小时。
* 产品负责人根据获得的信息，调整产品Backlog包。

## 回顾会议（Sprint Retrospective Meeting）

在每个Sprint结束后，Scrum团队会聚在一起开Sprint回顾会议，目的是回顾一下团队在流程人际关系以及工具方面做得如何。团队识别出哪些做得好，哪些做得不好，并找出潜在 的改进事项，为将来的改进制定计划。所有的Scrum会议都是限定时⻓长的，Sprint回顾会议的 推荐时⻓长是Sprint中的每一周对应一个小时(译者注:⽐比如，一个Sprint包含2个星期，则 Sprint回顾会议时⻓长为2个小时)。

Scrum团队总是在Scrum的框架内，改进他们自己的流程。

回顾会议过程如下：

1. Scrum Master介绍本Sprint的主要情况，导出度量表，进行项目数据分享。
2. 检查上次回顾会议行动计划的执行结果。
3. 每位项目成员总结三方面的内容：

* 哪些活动应该开始实施。
* 哪些活动应该停止实施。
* 哪些活动应该得到改进。

1. 对总结内容进行归结。
2. 对归结内容进行投票，每人最多3票。
3. 对票数较高的内容进行讨论，输出为行动计划，指定责任人。