

# Scrum 指南

---

Scrum 的权威指南：  
游戏规则



*Jeff Sutherland*



*Ken Schwaber*

*July 2011*

*由 Ken Schwaber 和 Jeff Sutherland 开发并维护*

# 目录

---

Scrum 指南的目的.....	3
Scrum 概述.....	3
Scrum 框架.....	3
Scrum 理论.....	4
Scrum .....	5
Scrum 团队.....	5
产品负责人 .....	5
开发团队 .....	6
Scrum Master .....	6
Scrum 事件.....	7
Sprint .....	7
Sprint 计划会议.....	8
每日例会 .....	9
Sprint 评审.....	10
Sprint 回顾.....	11
Scrum 工件.....	11
产品待办事项列表 .....	11
Sprint Backlog .....	13
增量 .....	13
“完成”的定义 .....	13
结论 .....	14
致谢 .....	15
人 .....	15
历史 .....	15
翻译 .....	15

## Scrum 指南的目的

Scrum 是用于开发和维持复杂产品的框架。这份指南包含了 Scrum 的定义，其中包括 Scrum 的角色、事件、工件，以及把它们绑在一起的规则。Ken Schwaber 和 Jeff Sutherland 开发了 Scrum，Scrum 指南也由他们撰写提供。他们是 Scrum 指南的后盾。

## Scrum 概述

Scrum: Scrum 是一个框架，在这个框架中人们可以应对复杂的适应性问题，同时也能颇有成效地和具有创造性地交付最高价值的产品。Scrum 是：

- 轻量级的
- 容易理解的
- 极难驾驭的

自从上世纪 90 年代初期，Scrum 框架就已经应用于处理复杂产品开发。Scrum 不是构建产品的一种过程或一项技术，而是一个框架，在这个框架里可以应用各种流程和技术。Scrum 能使产品管理和开发实践的相对功效显现出来，以便随时改进。

## Scrum 框架

Scrum 框架包括 Scrum 团队及其相关的角色、事件、工件和规则。框架中的每个模块都有一个特定的目的，对 Scrum 的成功和使用都至关重要。

使用 Scrum 框架的特定策略根据情况不同而不同，在这里不作介绍。

Scrum 的规则把事件、角色和工件绑在一起，控制着它们之间的关系和交互。Scrum 的规则会在这篇文档中通篇描述。

## Scrum 理论

Scrum 基于经验性过程控制理论，或者称为经验主义。经验主义主张知识源于经验，以及基于已知的东西做决定。Scrum 采用迭代、增量的方法来优化可预见性并控制风险。

Scrum 的三大支柱支撑起每个经验性过程控制的实现：透明性、检验和适应。

### 透明性

过程的重要方面必须对那些负责产出的人可见。透明性要求用通用的标准来定义那些方面，使观察者能对所见事物有共同理解。

例如：

- 所有的参与者必须有谈论流程的共同语言
- 完成工作的人和验收工作成果的人必须有共同的“完成”的定义。

### 检验

Scrum 使用者必须经常检验 Scrum 的工件和过程是否服务于目标，以确保及时发现不该存在的偏差。检验不必太频繁，以至于影响到工作本身。当熟练的检验员勤勉地检验工作重点时，效果最佳。

### 适应

如果检验员发现过程中的一个或多个方面背离了可接受标准，并且最终产品不合格时，就必须对过程或是材料进行调整。调整工作必须尽快实施以减少进一步的偏差。

Scrum 指定了 4 个检验和适应的正式时机，将在“Scrum 事件”中详细描述：

- Sprint 计划会议
- 每日例会
- Sprint 评审会议
- Sprint 回顾会议

## Scrum

Scrum 是一个支持复杂产品开发的框架，包含 Scrum 团队及其相关的角色、事件、工件和规则。框架中的每个模块都有一个特定的目的，对 Scrum 的成功使用至关重要。

Scrum 的规则把事件、角色和工件绑在一起，控制着它们之间的关系和交互。Scrum 的规则会在这篇文档中通篇描述。

## Scrum 团队

Scrum 团队包含产品负责人、开发团队和 Scrum Master。Scrum 团队是自组织且跨功能的。自组织团队选择如何最好地完成他们的工作，而不是由团队外的其他人来指使他们。跨功能团队拥有完成工作所需要的全部技能，不需要依赖团队外部的人。Scrum 团队模式的目的是最大限度地优化适应性、创造性和生产力。

Scrum 团队迭代增量交付产品，最大化反馈的机会。增量交付“完成”的产品保证了可工作产品的潜在可用版本总是存在。

## 产品负责人

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。

## 开发团队

开发团队包含了专业人员，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（即使是 Scrum Master 都不可以）告诉开发团队如何把产品代办事项列表变成潜在可发布的功能。
- 开发团队是跨功能的，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。

## 开发团队的规模

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 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。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 事件

## 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功能
- 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 事件
- 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 Service to the Organization

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
- 触发能增长 Scrum 团队生产力的改变
- 与其他 Scrum Master 一起工作，来增加组织中 Scrum 应用的效力

## Scrum 事件

Scrum 中指定了一些事件来创造规律性，也试图最小化对 Scrum 中未定义会议的需求。Scrum 中的事件是有时间盒限定的，也就是说每个事件都有时间限制的。这保证了花在计划上的时间是适当的，在计划的过程中没有浪费。

Sprint 是容纳所有其他事件的容器，除了其自身以外，Scrum 中的每个事件都是检验和适应的机会。这些事件被特别用来保证关键的透明性和检验。如果不能成功地包含这些事件中的任何一个，都会降低透明性，丧失检验和适应的机会。

## Sprint

Scrum 的核心是小于或者等于一个月时间盒的 Sprint，其产出是“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、每日例会、开发工作、Sprint 评审会议和 Sprint 回顾会议构成。

在 Sprint 中：

- 不能做出影响 Sprint 目标的改变
- 开发团队的组成和质量目标是不变的
- 随着知识的增加，产品负责人和开发团队可以澄清或者重新商谈开发范围

每个 Sprint 都可以被视为一个项目，范围不超过一个月。和项目一样，Sprint 被用来实现一些东西。每个 Sprint 都会定义要构建什么东西，一份设计和灵活的计划会指导构建、工作和最终结果。

Sprint 的范围被限制在一个月中。如果范围过大，对“要构建什么东西”的定义就会改变，复杂性会提升，风险也会增加。Sprint 通过确保至少每月一次对过程是否服务于目标的检验和适应来实现可预见性。Sprint 也把风险限制在一个月的成本上。

## 取消一个 Sprint

Sprint 可以在 Sprint 时间盒结束之前取消。只有产品负责人才有取消 Sprint 的权力，但他做这样的决定也可能是受到利益相关人、团队或是 Scrum Master 的影响。

如果某个 Sprint 的目标过时了，那么就需要取消该 Sprint。比如公司的发展方向，或是市场、技术等情况发生了变化，这些变化都可能导致取消 Sprint。总的来说，如果某个 Sprint 对于其所在环境来说失去了价值和意义，那么它就应该被取消。然而，因为 Sprint 周期都较短，所以很少发生取消 Sprint 的情况。

当某个 Sprint 被取消时，任何做完和“完成”的产品待办事项列表条目都需要评审。假如有些条目已经潜在可交付，那产品负责人就会采纳它。所有未完成的条目就都要放回到产品待办事项列表中，并重新估算。花在它们身上的工作会迅速贬值，所以需要频繁地重估。

取消 Sprint 会消耗资源，因为每个人都需要在新召开的 Sprint 计划会议中重新开始另一个 Sprint。Sprint 终止通常会对团队造成重创，因此这种情况也非常少见。

## Sprint 计划会议

Sprint 中要完成的工作会在 Sprint 计划会议中计划。这份计划是由整个 Scrum 团队共同协作完成的。

对于周期为一个月的 Sprint，计划会议的时间盒限定为 8 小时。对于较短的 Sprint，根据 Sprint 的长度，按比例缩小会议的时间。比如，两周的 Sprint 对应 4 小时的 Sprint 计划会议。

Sprint 计划会议分为两部分，每一部分都占用 Sprint 计划会议时间盒长度的一半。

两部分的计划会议分别回答以下两个问题：

- 下个 Sprint 中将交付什么增量结果？
- 要交付增量需完成什么样的工作？

### 第一部分：这个 Sprint 要做什么？

在这一部分中，开发团队预计这个 Sprint 中将要开发的功能。产品负责人向开发团队呈现排列好的产品代办事项列表条目，然后整个 Scrum 团队共同协作理解 Sprint 中的工作。



Sprint 会议的输入是：产品待办事项列表、最新的产品增量、开发团队在这个 Sprint 中的接受力和以往的表现。开发团队自己决定选择待办事项列表条目的数量，因为只有开发团队可以评估在接下来的 Sprint 内可以完成什么工作。

在预计这个 Sprint 中可交付的产品待办事项列表条目后，Scrum 团队会打造一个 Sprint 目标。Sprint 目标是在这个 Sprint 要达到的目的，通过实现产品待办事项列表来达到。它也为开发团队提供指导，使团队明确构建增量的目的。

## 第二部分：选出的工作如何完成？

选出 Sprint 的工作之后，开发团队决定如何在 Sprint 中把这些功能构建成“完成”的产品增量。这个 Sprint 中所选出的产品代办事项列表条目以及交付它们的计划被称为 Sprint 代办事项列表。

开发团队通常先由系统设计开始，并设计把产品代办事项列表转换成可工作的产品增量所需要的工作。工作的大小或预估的工作量可以不同。然而，在 Sprint 计划会议中，开发团队需要做足够的计划来预计他们在即将到来的 Sprint 中所能完成的工作。开发团队所计划的 Sprint 最初几天的工作必须在会议结束前分解为一天或少于一天的量。开发团队自组织地领取 Sprint 代办事项列表中的工作，可以在 Sprint 计划会议中，也可以在 Sprint 的过程中按需领取。

产品负责人会参加 Sprint 计划会议的第二部分，对选定的产品待办事项列表条目做出澄清，并协助团队权衡取舍。如果团队认为工作量过大或太小，就可以和产品负责人重新协商 Sprint 待办事项列表条目。开发团队也可以邀请其他人员参加会议，以寻求技术和领域建议。

Sprint 计划会议结束时，开发团队应该能够向产品负责人和 ScrumMaster 解释他们将如何以自组织团队的形式完成 Sprint 的目标并创造期望的产品增量。

## Sprint 目标

Sprint 目标为开发团队在 Sprint 中所实现的功能留有一定的弹性。

在开发团队的工作进程中，必须时刻牢记目标。为了达成 Sprint 目标，需要实现功能和技术。如果工作和预期的不同，开发团队需要与产品负责人协同商议本个 Sprint 中 Sprint 代办事项列表的范围。

Sprint 目标可能是更远大的产品路线图中的一个里程碑。

## 每日例会

每日例会是 15 分钟时间盒的开发团队事件，为的是同步活动并创建下个 24 小时的计划。这需要检验上个每日例会以来的工作和预测下个每日例会之前所能完成的工作。

每日例会在同一时间，同一地点进行，来降低复杂度。会议上，每个团队成员需要汇报以下三个问题：

- 从上次会议到现在都完成了哪些工作；
- 下次每日例会之前准备完成什么；
- 工作中遇到了哪些障碍。

开发团队用每日例会来评估完成 Sprint 目标的进度，并评估完成 Sprint 代办事项列表的进度趋势。每日例会优化开发团队达成 Sprint 目标的可能性。开发团队经常在每日例会后立即重新计划 Sprint 中的剩余工作。每天，开发团队应该能够向产品负责人和 Scrum Master 解释他们将要如何一起工作成一个自组织团队来达到目标并在剩余的 Sprint 中创造期望的产品增量。

Scrum Master 确保开发团队每日例会如期举行，开发团队自己则负责召开会议。Scrum Master 指导团队把会议控制在 15 分钟的时间盒内。

Scrum Master 强制每日例会的规则，只有开发团队成员才能参加。每日例会不是进度汇报会议，只是为将产品待办事项列表条目转化成为增量的人（团队）而召开的。

每日例会可以增强交流沟通、省略其他会议、确定并排除开发遇到的障碍、强调和提倡快速决策、提高每个成员对项目的认知程度。这是关键的检验和适应的会议。

## Sprint 评审

Sprint 评审会议在 Sprint 的结尾召开，用以检验所交付的产品增量并按需调整产品代办事项列表。在 Sprint 评审会议中，Scrum 团队和干系人沟通 Sprint 中完成了哪些工作。然后，根据完成情况和 Sprint 期间产品待办事项列表的变化，与会人员确定接下来的工作。这是一个非正式会议，会议中进行增量演示，以引发反馈并促进合作。

一个月的 Sprint 通常对应 4 小时时间盒的评审会议。对于时间少于一个月的 Sprint 来说，会议的长度会根据 Sprint 的长度按比例缩减。比如，两周的 Sprint 对应 2 小时的 Sprint 评审会议。

Sprint 评审会议包含以下因素：

- 产品负责人确定哪些工作“完成”了，哪些工作没有“完成”
- 开发团队讨论在 Sprint 中哪些工作进展顺利、遇到了什么问题、问题是如何解决的
- 开发团队演示“完成”的工作并解答关于所交付增量的问题
- 产品负责人和与会人员按现实情况讨论产品待办事项列表，并基于当前的进度推测可能的完成日期
- 整个团体就下一步的工作进行探讨，这样，Sprint 评审会议就能为接下来的 Sprint 计划会议提供了有价值的信息。

Sprint 回顾的结果是一份修订的产品代办事项列表，确定了很可能进入下个 Sprint 的产品代办事项列表条目。也有可能为了应对新机遇而全局调整产品代办事项列表。

## Sprint 回顾

Sprint 回顾会议是 Scrum 团队检验自身并创建下个 Sprint 改进计划的机会。

Sprint 回顾会议发生在 Sprint 评审会议结束之后和下一个 Sprint 的计划会议之前。对于长度为一个月的 Sprint，这是一个 3 小时时间盒的会议。对于较短的 Sprint，按比例缩短会议的时间。

Sprint 回顾会议的目的是：

- 对前一个 Sprint 周期中的人、关系、过程和工具进行检验
- 识别并排序做得好的和能潜在改进的主要条目
- 创建实施改进 Scrum 团队工作方式的计划

Scrum Master 鼓励团队在 Scrum 的过程框架内改进开发过程和开发实践，使得他们能在下个 Sprint 中更高效更愉快。在每个 Sprint 回顾会议中，Scrum 团队通过适当调整“完成”的定义，来计划提高产品质量的方法。

在 Sprint 回顾会议结束时，Scrum 团队应该确定了下个 Sprint 中需要实施的改进。在下个 Sprint 中实施这些改进是基于 Scrum 团队对自己的检查而做出的适应。虽然改进可以在任何时间执行，Sprint 评审会议提供了一个专注于检验和适应的活动。

## Scrum 工件

Scrum 的工件以不同的方式表现工作和价值，可以用来提供透明性以及检验和适应的机会。Scrum 中所定义的工件能最大化关键信息的透明性，来保证 Scrum 团队成功地交付“完成”的增量。

### 产品待办事项列表

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。

产品待办事项列表永远是不完全的，最初的版本只列出最初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。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。

“产品代表事项列表优化”是增添细节、估算和排序条目的举动。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优化的时候，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。

优化在 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。优化通常不占用超过开发团队 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。

## 监控向目标前进的进度

在任何时间，达成目标的剩余工作量是可以被累计的。产品负责人至少在每个 Sprint 评审的时候追踪剩余工作总量。产品负责人把这个数量与之前 Sprint 评审时的剩余工作量做比较，来评估在希望的时间点完成预计工作达成目标的进度。这份信息对所有的干系人都透明。

Scrum 不考虑已经花在产品代办事项列表条目上的工作时间。我们只关心剩余工作和日期这两个变量。

各种趋势燃尽图、燃烧图和其他计划实践都能用来预测进度。它们已经被证实有用。然而，这并不能代替经验主义的重要性。在复杂的环境下，将要发生的事情是未知的，只有已经发生的事情才能用来做前瞻式的决策。

## 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一个增量中，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。

## 监控 Sprint 进度

在 Sprint 中的任意时间点，Sprint 待办事项列表的所有剩余工作总和都可以被计算。开发团队至少在每日例会时追踪所有的剩余工作。开发团队每天追踪剩余总和并预测达成 Sprint 目标的可能性。通过在 Sprint 中不断追踪剩余工作，开发团队可以管理自己的进度。

Scrum 不考虑已经花在 Sprint 待办事项列表上的工作时间。我们只关心剩余工作和日期这两个变量。

## 增量

增量是一个 Sprint 及以前所有 Sprint 中完成的所有产品代办事项列表条目的总和。在 Sprint 的结尾，新的增量必须“完成”，这意味着它必须可用并且达到了 Scrum 团队“完成”的定义的标准。无论产品负责人是否决定真正发布它，增量必须可用。

## “完成”的定义

当产品代办事项列表条目或者增量被描述为“完成”的时候，每个人都必须理解“完成”意味着什么。虽然这在不同的 Scrum 团队之间会有巨大的差别，但是团队成员必须对完成工作意味着什么有相同的理解，这样才能保证透明性。这就是 Scrum 团队的“完成”定义，用来评估产品增量在什么时候完成。

这个定义也同时被用来指导开发团队了解在 Sprint 计划会议的时候他们能选择多少产品代办事项列表条目。每个 Sprint 的目标都是交付遵循 Scrum 团队当前的“完成”定义的潜在可交付功能增量。

开发团队在每个 Sprint 交付产品功能增量。这个增量是可用的，所以产品负责人可以选择立即发布它。每个增量都附加于之前所有增量并经过充分测试，以此保证所有的增量都能工作。

随着 Scrum 团队的成熟，我们预期“完成”的定义会扩大，包含更严厉标准来保证高质量。

## 结论

Scrum 是免费的，在这份指南中提供。Scrum 的角色、工件、事件和规则是不可变的。虽然只实施部分的 Scrum 是可能的，但这样就不是 Scrum 了。Scrum 只以整体的形式存在，才能作为其他技术、方法论和实践的容器良好运作。



## 致谢

### 人

在千千万万对 Scrum 做出贡献的人中，我们要特别感谢那些在其最初 10 年提供帮助的人们。首先，要提到 Jeff Sutherland 及与之工作的 Jeff McKenna 和 Ken Schwaber 还有 Mike Smith 和 Chris Martin。许多人在随后的几年中也都做出了贡献，没有他们的帮助，Scrum 不会如今天这样精致。David Starr 在制定这个版本的 Scrum 指南时，提供了关键的见解和编辑技巧。

### 历史

Ken Schwaber 和 Jeff Sutherland 在 1995 年的 OOPSLA 大会中首次共同演说 Scrum。那次演讲本质上记录了 Ken 和 Jeff 在之前几年使用 Scrum 时所学到的经验。

Scrum 的历史已经算是很长了。我们对首批尝试和提炼 Scrum 的公司：Individual, Inc.、Fidelity Investments 和 IDX（现在的 GE 医疗）表示致敬。

### 翻译

这份指南是从 Ken Schwaber 和 Jeff Sutherland 的英文原版翻译而来。参与翻译的包括鲍央舟和孙媛。

## 版本信息

2011年7月的Scrum指南与之前2010年2月的版本不同。我们特意试图把技术和最佳实践从Scrum的核心中去除。它们会随着环境的不同而不同。我们之后会新写一份“最佳实践”的纲要来提供我们的经验。

Scrum指南记录了Jeff Sutherland和Ken Schwaber开发和维持Scrum 20多年来的历史。其他的资源提供了模式、流程，以及关于实践、推动力和工具如何补充Scrum框架的深入见解。他们优化了生产力、价值、创造力和自豪感。

版本说明涵盖了此版本与2010年2月份版本的以下不同点，也会在别的地方发布：

1. 发布计划
2. 发布燃尽图
3. Sprint Backlog
4. 产品和Sprint Backlog燃尽图
5. “承诺”（commit）现在改成“预测”（forecast）
6. “团队”改成“开发团队”
7. 鸡和猪……Scrum的传统故事
8. “排序”（ordered）而不是“优先级次序”（prioritized）