



Scrum理论与实践的轻量级指南 第2.0版

Pete Deemer
GoodAgile
www.goodagile.com

Gabrielle Benefield
Evolve
www.evolvebeyond.com

Craig Larman
www.craiglarman.com

Bas Vodde
Odd-e
www.odd-e.com

孙媛, 尹哲 译



InfoQ
ENTERPRISE SOFTWARE
DEVELOPMENT SERIES

读者提示：在网络上可以找到很多简明的Scrum描述，这篇简介旨在提供更多在实践方面的细节。它并无意成为Scrum学习的终点。建议考虑要引入Scrum的团队用Ken Schwaber所著的《Scrum敏捷项目管理实战》，或者Mike Cohn所著的《Scrum敏捷软件开发》来装备自己，并且充分利用众多优秀的Scrum培训和教练等资源，具体细节可以在scrumalliance.org找到。我们感谢Ken Schwaber, Jeff Sutherland博士以及Mike Cohn的无私贡献。

本简章的最新英文版本在：http://www.infoq.com/minibooks/Scrum_Primer

翻译版本在：<http://www.scrumprimer.org/>

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

超越传统开发方式

传统的开发方式由单一职能团体组成，反馈环延迟甚至薄弱，使用预言性质的计划和从分析到测试流程，这在变化无常的当今世界中成效甚微。由于直到开发后期才会有真正可工作的软件，这种方式会延迟反馈、学习，以及潜在投资回报。从而导致透明度不足、改进能力缺失、灵活度减少、商业和技术风险增加。

取而代之的是跨职能团队与迭代开发，这种方式也存在了几十年，但并不如传统模式那样被广泛使用。

Scrum把已被证实的产品开发概念打包到简单的框架中，包括：真正的团队、跨职能团队、自我管理团队、短迭代全周期反馈环、降低变动成本。这些概念提升了敏捷性与反馈、使提早实现投资回报成为可能并降低风险。

概括

Scrum是跨职能团队以迭代、增量的方式开发产品或项目的一种开发框架。它把开发组织成被称为Sprint的工作周期。这些迭代每个都不超过4周（最常见的是两周），并且无间歇地相继进行。Sprint是受时间箱限制的，无论工作完成与否它们都会在特定日期结束，并且从不延长。通常由Scrum团队来选定一个Sprint的时长，并且对于他们所有的Sprint都使用这一时长，直到这个团队能力提升，可以使用较短周期。在每个Sprint的初始，跨职能团队（大约7名成员）从排好优先级的列表中选择事项（客户需求）。团队对于在Sprint结尾他们相信自己可以交付哪些目标集合达成一致意见，这些交付应该是有形的并且能被真正“完成”的。在Sprint过程中不可以增加新事项，Scrum在下一Sprint时才接受变化，当前这么短的一个Sprint周期里只注重于短小、清晰、相对固定的目标。团队每天都进行简短会面来检验工作进程，并调整后续步骤以确保完成剩余工作。在Sprint结尾，团队与利益关系人一起回顾这个Sprint，并演示所构建的产品。团队成员从中获取可以结合到下一Sprint中的反馈。Scrum强调在Sprint结尾产生真正“完成”了的可工作产品。在软件领域是指已经集成的、完全测试过的、已经为最终用户生成文档的、潜在可交付的系统。其中的主要角色、工件和事件如图表1所总结。

SCRUM

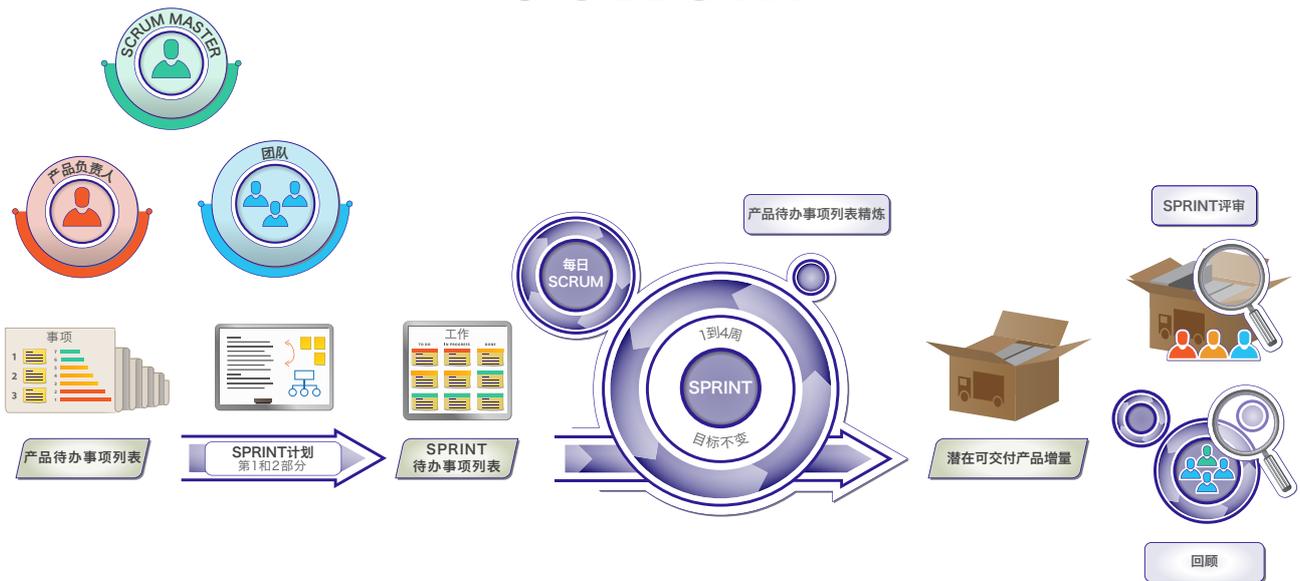


图1 Scrum概括

Scrum中的一大主题就是“审视并调整”。因为开发工作不可避免地包含学习、创新和意外事件，Scrum强调进行小步骤开发，同时检验最终产品和当前实践的功绩，然后调整产品目标与过程实践。周而复始。

Scrum中的角色

在Scrum中有三个角色：产品负责人，团队和ScrumMaster。他们在一起被称为Scrum团队。

产品负责人负责最大化投资回报（ROI，return on investment），通过确定产品特性，把它们翻译成有优先级的列表，为下一个Sprint决定在这个列表中哪些应当优先级最高，并且不断地重新调整优先级和梳理这个列表。对于商业产品而言，产品负责人对产品的盈亏负责。对于内部应用的情况，从非（可以产生收入的）商业产品的角度来讲产品负责人不为ROI负责，但从选择每个Sprint具有最高价值事项的角度来讲他们仍为最大化投资回报负责。在实践中，“价值”的含意很模糊，优先级的制定可能会受希望让关键客户满意、保持与战略目标一致、解决风险、改进以及其他因素的影响。在有些情况下，产品负责人与客户是同一个人。这在内部应用时很常见。有时，客户可能是带着不同需求的几百万人，在这种情况下，产品负责人与很多产品开发组织中的产品经理或者产品市场经理类似。然而，产品负责人与传统的产品经理有些不同，因为产品负责人积极地并经常地与团队互动，重点放在与所有的利益相关人一起工作并且评审每个Sprint的结果，而不是把开发相关的决定交由项目经理来代理。值得注意的是，在Scrum中，有一个人，且只有一个人做产品负责人并拥有其最终的权力。他对工作所产生的价值负责。然而这个人却不一定必须独自工作。

团队（也叫开发团队）建造产品负责人所指定的产品，例如应用程序或者网站。Scrum中的团队是“跨职能”的，它包含了每个Sprint为了交付潜在可交付的产品所需的所有专业能力，并且它是“自组织”（自我管理）的，被给予很高程度的自治和责任。由团队来决定在某个Sprint中要完成多少（从产品负责人给出的集合中的）事项，以及怎样才能最好地达到这个目的。

团队中的每个成员都只是“团队成员”。请注意在采用Scrum的团体中没有任何固定的专业头衔。不会有业务分析员，没有数据管理员，没有架构师，没有团队组长，没有交互/用户体验设计师，也没有程序员，他们在每个Sprint中以任何恰当的方式一起工作来达到他们为自己设置的目标。

因为只有“团队成员”，这种团队不只是跨职能的而且还会表现出“多重学习”的特点。每个人当然都有特长，但仍继续学习其他专长。每个人都会有主要的、次要的甚至是第三位的技能，并准备好“到工作所需要的地方去”。每个人都可能承担自己并不熟悉领域的任务来帮助完成一个事项。例如，一个主要技能是交互设计的人可能以自动化测试为次要技能。以技术文档写作为主要技能的人可能同时帮助做分析和编程。

Scrum中的团队由七个人左右（加上或减去两个人）组成，对于一个软件产品来讲团队可能包含具有分析、开发、测试、接口设计、数据库设计、架构、文档等等技能的人。团队开发产品，并且向产品负责人提供如何把产品做得更出色的想法。在Scrum中，当所有成员在Sprint中都百分之百地投身于同一个产品的工作时，产量和效率最高。团队避免了在多个产品或项目之间的多任务，从而避免了分散注意力和工作内容切换的高代价消耗。稳定的团队会具有更高的生产率，因此要避免改变团队成员。有很多人组成的产品开发团体要被组织成多个团队，每一个都关注于产品的不同特性，紧密合作，共同努力。由于一个团队往往要做所有完成以用户为中心的特性所需的工作（计划、分析、编程和测试），这样的团队也被称为“特性团队”。

ScrumMaster帮助产品开发团体学习并应用Scrum来达成商业价值。ScrumMaster会做任何力所能及的事情来帮助团队、产品负责人和组织取得成功。ScrumMaster不是团队成员的经理，也不是项目经理、团队带头人或者团队的代表。相反，ScrumMaster为团队服务。他帮助移除阻碍，保护团队免受外部干扰，并且帮助团队采用现代的开发实践。他教育、培养并指导产品负责人、团队以及组织中其他的人来提高他们使用Scrum的技巧。ScrumMaster是教练和导师。ScrumMaster确保每个人（包括产品负责人和管理者）理解Scrum中的原则和实践，并且他们帮助带领组织完成敏捷开发要取得成功所需的那些通常都很困难的改变。因为Scrum使得影响团队和产品负责人有效性的很多阻碍和风险都变得可见了，由专注的ScrumMaster全身心地帮助解决这些问题就非常重要，否则团队或者产品负责人会发现很难取得成功。ScrumMaster应该是专门全职的，然而，小一点的团队可能由其中一个团队成员扮演这个角色（这样做时他们只能肩负较轻的日常工作量）。出色的ScrumMaster可能来自于任何背景或专业：工程、设计、测试、产品管理、项目管理或者质量管理。

ScrumMaster和产品负责人不可以是同一个人，因为他们的关注点太过不同，合并这两个角色通常导致困惑和冲突。合并这两个角色的一个常见的不幸结果是产品负责人进行微观管理，这与Scrum所要求的自我管理团队相冲突。与传统的管理者不同，ScrumMaster不会告诉别人要做什么或者分配任务，他们推进流程，为团队的自组织和管理提供支持。如果ScrumMaster以前在管理团队的岗位上工作，那么要在Scrum上取得成功，他需要很大程度上改变与团队互动的心态和思考方法。

注意：在Scrum中完全没有项目经理这个角色。这是因为不需要。项目经理的传统职责已经被分开并分配在Scrum的三个角色中了，主要是给了团队和产品负责人，而不是ScrumMaster。在实践Scrum时加入一个项目经理意味着对Scrum基本的理解错误，典型的结果是责任冲突、职权不明和局部优化。可能一个（从前的）项目经理会担当ScrumMaster的角色，但这能否成功很大程度上取决于个人，以及这个人对这两个角色基本区别的理解程度，不光是日常职责，还有取得成功所需的思维方式。一个好的、能够透彻地了解ScrumMaster这个角色，并开始培养取得成功所需的核心技能的方式是参加由Scrum联盟所提供的认证ScrumMaster培训。

除了这三种角色，还有其他的利益相关人也会为产品的成功做出贡献，包括管理者，客户和最终使用者。有些利益相关人，如职能性经理（例如工程经理）可能会发现他们的角色仍有价值但在引入Scrum时发生了改变。例如：

- 他们要以尊重Scrum的规则和精神的方式来为团队提供支持
- 他们帮助移除团队和产品负责人发现的阻碍
- 他们要确保他们的专业和经验可以被利用

在Scrum中，这些个人把花在扮演“保姆”角色上的时间（分配任务、得到状态报告以及其他各种形式的微观管理）替换为扮演团队的“导师”和“仆人”（指导、培养、帮助移除障碍，帮助解决问题，提供有创造性的输入以及指导团队成员的技能发展等）。在这种转换中，管理者们可能需要改变他们的管理风格。例如，使用启发式的问题来帮助团队发现问题的解决方案，而不是简单的决定一个方案然后分配给团队去做。

产品待办事项列表（Product Backlog）

当一个团体计划向Scrum过渡时，在首个Sprint可以开始之前，他们需要有一个（按1、2、3顺序）排好优先级的、以客户为中心的特性列表，即产品待办事项列表。

产品待办事项列表存在（并演化）于产品的整个生命周期，它是产品的路线图（如图2和3）。任何时候，产品待办事项列表都是“团队依照优先排列顺序完成工作”的唯一、最终的概括。一个产品只有一个产品待办事项列表，这就意味着产品负责人必须纵观全局做出优先级排列的决策，以体现利益相关人（包括团队）的意愿。

优先级	事项	细节 (wiki链接)	初始规模 估算	每个Sprint 的新估算					
				1	2	3	4	5	6
1	作为买家，我想把书放入购物车（见wiki页面用户界面草图）	...	5						
2	作为买家，我想从购物车中删除书	...	2						
3	提高交易处理性能（见wiki页面目标性能指标）	...	13						
4	探讨加速信用卡验证的解决方法（见wiki页面目标性能指标）	...	20						
5	将所有服务器升级到Apache 2.2.3	...	13						
6	分析并修复订单处理脚本错误（错误号：14834）	...	3						
7	作为购物者，我想创建并保存愿望表	...	40						
8	作为购物者，我想增加或删除愿望表中的条目	...	20						

图2 产品待办事项列表



图3 可视化管理：将产品待办事项列表事项贴在墙上

产品待办事项列表包括不同种类的事项：全新的客户特性（“使所有用户可以将书籍放入购物车”），也有主要的技术改进目标（如“用Java代替C++重写系统”），还有改进目标（如“加速测试”）、调研工作（“探讨加速信用卡有效验证的解决方法”），还可能会有已知的缺陷（“分析并修复订单处理脚本错误”），如果问题不多的话（如果系统有很多缺陷，通常就会有单独的缺陷跟踪系统）。

产品待办事项列表事项可以用任何明确且可持续的方式来表述。与常见的误解正相反，产品待办事项列表并不是由“用户故事”组成的，而是只包含事项。这些事项可以通过用户故事、用例或任何大家认为有用的需求方式来体现。但是无论何种方式，多数事项应当以交付客户价值为重点。

一个好的产品待办事项列表要做到DEEP.....

粗细适宜的（Detailed appropriately）。优先级列表顶端的事项比低级别的事项更为精确和详细，因为前者要比后者先被开发。比如，待办事项列表顶端的百分之十可能包含非常小且分析得很详细的事项，而其他的百分之九十则不是那么具体。

估算过的（Estimated）。当前发布版本的事项需要有估算，而且随着大家了解得更多和新信息的融入应当在每个Sprint中重新考虑这些估算。团队提供给产品负责人产品待办事项列表中每个事项的工作量估算和技术风险估算。产品负责人和其他商业利益相关人提供产品需求的价值信息，其中可能会包括获得的收益、减少的成本、商业风险、对不同利益相关人的重要性等等。

涌现式的（Emergent）。为了响应学习和变化，要定期梳理产品待办事项列表。每个Sprint，可能要加入、删除、修改、分解或者调整事项的优先级别。因此，产品负责人会不断地更新产品待办事项列表，以反映客户需求的变化、新想法或见解、竞争而导致的变化、出现的技术障碍等等。

排好优先级的（Prioritized）。在产品待办事项列表顶端的事项具有最高优先级，或者是从1开始顺序排列。一般来说，最高优先级别的事项应当最物超所值：高收益（商业价值）低花销（成本）。提高某事项优先级的另一诱因是在高风险来袭之前及早解决掉它。

传统的开发方式通常不强调依照最物超所值的标准交付，但这是Scrum的主题之一，因此产品负责人就必须掌握如何评定“商业价值”。这可能需要ScrumMaster协助产品负责人来学习。“商业价值”到底意味什么呢？一些产品团体对于每个产品待办事项列表事项使用简单、相对的价值点估算，这些就综合成了对包括获得收益、成本减少、利益相关人偏好、市场区分等等因素的“推测”。一些团体更是由一个或几个客户担负费用来资助某特定事项的开发，然后用该事项的确切（短期）收益来代表价值。对于另一些团体，这种针对特定事项的价值估算太不集中或太细碎，他们使用范围更大的、基于商业成果的方式（“在九月一日之前增加百分之十的订购”），这样价值只有在当多个对产生成果起作用的事项一起交付的时候才会实现。这种情况下，产品负责人需要确定最小可行产品（Minimum Viable Product）的下一个增量。

关于工作量估算，通常的做法是按照相对规模（影响工作量的因素、复杂程度及不确定程度）使用“故事点”或简单的“点”为单元来估算。

这些都只是一些建议，Scrum并没有规定在产品待办事项列表中对事项表述或排列优先级的方法，也没有规定估算的方法或单位。

在Scrum中有一个常见的做法是追踪每个Sprint完成的工作量，比如每个Sprint平均完成26点。如果保持平均水平并且没有其他变化的条件下，根据这个信息他们可以预测出完成所有特性的发布日期，或到特定日期可以完成多少特性。这个平均值就称为“速率”。速率与产品待办事项列表事项规模估算用相同的单位来表达。

产品待办事项列表中的事项在规模或工作量上可以有非常大的不同。较大事项会在产品待办事项列表梳理会或Sprint计划会议上被分解为若干个较小事项，较小的事项也可能被合并。接下来几个Sprint的产品待办事项列表事项应当是很小且足够精细的，那么团队可以做到充分理解，使在Sprint计划会议上做出的预期更有实际意义，这就是“拿来就能做”的规模。

耗费很多时间和资金的重大工程改进应当被包含到产品待办事项列表中，因为它们很有可能是潜在的商业投资，最终由以商业为导向的产品负责人来决定。注意，在Scrum中，团队对于从产品待办事项列表中提取多少事项到当前Sprint中有自主的权力，因此他们可以独立自主地进行较小的工程改进工作，这些工作会被纳入到正常商业运转成本中，而且这些也是对开发人员妥善完成工作的要求。这就是说在每个Sprint中，团队的大部分时间通常应当放在产品负责人目标上，而不是内部工程任务。

对Scrum的一个误解是说它避免撰写详细规格说明书，实际上，产品负责人和团队决定他们需要多少细节而且每个待办事项列表事项的要求都不同，这些都依据团队的见解和其他一些因素来决定。要用最少的空间描述最重要的东西，换句话说就是不要描述事项的每个细节，只要清晰明了可以理解，并随着团队、产品负责人和利益相关人之间的不断对话而扩充。低优先级的产品待办事项列表事项，它们可能不会在近期内被纳入工作，因此它们通常是“粗粒度”（大的、需求并不详细的）的。高优先级、细粒度的产品待办事项列表事项将会在近期内被实现，它们的描述会更为详细。

完成的定义

每个Sprint的输出的正式名字为“潜在可交付产品增量”。在开始第一个Sprint之前，产品负责人、团队和ScrumMaster必须审视对于把一个产品待办事项列表中的事项做到潜在可交付所需要的所有事情。所有为了交付产品所需的活动都应被包含在“潜在可交付”的定义中，并且要在这个Sprint中完成。

遗憾的是，当团队开始使用Scrum时他们通常做不到在每个Sprint都能交付出“潜在可交付增量”这个目标。这通常是因为团队缺少自动化或者不够跨职能（例如，技术文档撰写者还没有被包含在跨职能团队中）。随着时间的过去，团队必须要提高从而能够在每个Sprint交付“潜在可交付产品增量”。但是为了开始，他们需要建立一个他们当前能力的基线。这会被记录在“完成的定义”中。

在第一个Sprint开始前，产品负责人和团队要对于“完成的定义”达成共识，“完成的定义”是创建“潜在可交付产品增量”所需要的活动的子集（对于一个好的团队来讲两者是一样的）。团队会根据“完成的定义”来计划他们在Sprint中的工作。

一个好的产品负责人总是会希望“完成的定义”与“潜在可交付”越接近越好，因为这样会增加开发中的透明度并降低延迟和风险。如果“完成的定义”不等同于“潜在可交付”，那么就会有工作被延迟到发布之前，这会导致风险和延迟。因此被延迟的工作有时被称为未完成的工作。

Scrum团队应当持续地改进，这一点会表现在对“完成的定义”的扩展上。

Sprint计划

概要：这是一个为Sprint做准备的会议，通常分成两部分（第一部分关于“做什么”，第二部分关于“怎么做”）。

参与者：第一部分：产品负责人、团队、ScrumMaster。第二部分：团队、ScrumMaster、产品负责人（不强制参加，但是要能被找到来回答问题）。

长度：每个部分的时间箱的小时数与Sprint的周数相等。

在每个Sprint的开始，会召开“Sprint计划会议”。它分为两个不同的子会议，其中第一个被称为“Sprint计划会议第一部分”。

在Sprint计划会议第一部分中，产品负责人和团队审视产品待办事项列表中产品负责人有兴趣在这个Sprint中实现的那些高优先级的事项。通常，这些事项应当已经在前一个Sprint中（的产品待办事项列表梳理会议里）被很好地分析过了，因此在这个会上只有较少的需要在最后时刻澄清的问题。在这个会议中，产品负责人和团队讨论产品待办事项列表中这些高优先级事项的目标和上下关联，让团队洞悉产品负责人的思想。第一部分关注于理解产品负责人想要什么以及为什么需要它。在第一部分的最后（总是很忙碌的）产品负责人可以离开，不过他必须在会议的第二部分过程中可以被联系到（例如，可以接电话）。

在第一部分中，团队和产品负责人还可以设计出“Sprint目标”。这是对Sprint目的的一个概括性的描述，在理想的情况下这些目的有一致的主题。Sprint目标也为团队实际可能的交付提供了范围上的灵活性，因为尽管他们可能不得不拿掉一些事项（因为Sprint是受时间箱控制的），他们却仍可以承诺交付与Sprint目标精神一致的、有形的并“完成”的一些东西。

Sprint中所承接的这些事项应当有多大？每个事项应当被拆得足够小，它在估计上只需要花比整个Sprint少得多的时间。一个常用的指导原则是一个事项从估计上来讲要小得足够由整个团队在四分之一Sprint或者更少的时间内完成。

Sprint计划会议第二部分关注于如何实现团队决定要开始做的那些事项。团队预期他们在Sprint结尾能够完成多少事项，从产品待办事项列表的顶部开始（换句话说，就是从对于产品负责人来讲最高优先级的事项开始），并依次查看事项。以下是Scrum中的关键实践：由团队决定将完成多少工作，而不是由产品负责人分配给他们。因为团队是基于他们自己的分析和计划，这使得预期更可靠。虽然产品负责人对于团队接受多少工作没有控制权，但他明白产品待办事项列表中的事项是从顶部开始拿取的，换句话说，就是先拿那些被他评为重要的事项。团队可以为列表中很后面的事项进行游说，这通常发生在团队和产品负责人发现低优先级的某些东西很容易并可以恰当地与高优先级的事项合并时。

Sprint计划会议通常会持续几个小时，但对于两周的Sprint来讲不应超过四个小时，团队对于要完成的工作做出正式的预期，要做到这一点需要仔细的思考。第一部分和第二部分的时间箱长度是相等的。对于一个两周的Sprint来讲每个部分最多是两个小时。

Scrum并没有定义具体如何做Sprint计划会议第二部分。有些团队用他们以前Sprint的速率来指导可预期多少工作。有些团队会用先计算可容纳工作量的更精细的方式。

当使用可容纳工作量的方式时，团队的Sprint计划会议第二部分从估算每个成员有多少时间花在Sprint相关的工作上开始。换言之，就是他们的平均工作日减去花在参加会议、收发邮件、午餐休息等等时间。对于大多数人来讲这样的结果是平均每天4-6小时在Sprint相关的工作上，其他的时间花在了邮件、午餐、社交网站、开会和喝咖啡上。这就是团队接下来Sprint可容纳的工作量。算出了这个容量以后，团队要算出他们在这些时间里能完成产品待办事项列表中的多少事项，以及团队将如何完成它们。这通常从在白板上讨论设计开始。一旦大家理解了整体的设计，团队会把产品待办事项列表中的事项分解成较细粒度的工作。在开始处理产品待办事项列表中的事项以前，团队可能会先关注于为前一个Sprint的回顾会议中所创建的改进目标生成一些任务。然后，团队选择产品待办事项列表中的第一个事项，即对产品负责人来讲最高优先级的事项，然后依次处理，直到他们“容量填满”为止。他们为每个事项建立一个工作列表，有时由产品待办事项列表中的事项分解出的任务组成，或者当产品待办事项列表中的事项很小，只要几个小时就能实现时，就简单的由产品待办事项列表事项组成。这个在Sprint过程中要被完成的工作的列表叫做“Sprint待办事项列表”（参见图4和图5）。

		每日结束时所剩余工作量的最新估计							
产品待办事项列表事项	Sprint中的任务	志愿者	初始工作量估计	1	2	3	4	5	6
做为买家，我希望把书放到购物车中	修改数据库		5						
	创建网页(UI)		8						
	创建网页 (Javascript逻辑)		13						
	写自动化验收测试		13						
	更新买家帮助网页		3						
	...								
改进事务处理效率	合并DCP代码并完成分层测试		5						
	完成pRank的机器顺序		8						
	把DCP和读入器改为用pRank http API		13						

图4：一种建立Sprint待办事项列表的例子

在Sprint计划会议的结尾，团队为他们相信自己在Sprint结束时所能交付的工作设置一个现实的目标。过去，这被称为Sprint的承诺，团队承诺尽他们的最大努力来达到他们的目标。遗憾的是，这有时会被误读为血书一样的诺言，而不是团队认真地“决意做到”。为了避免这个混淆，现在Sprint的目标被称做“预期”，用来与产品负责人沟通。

Scrum鼓励多面手，而不只是“在其位，谋其政”，例如“测试人员”只做测试。换言之，团队成员“工作需要什么就做什么”并且尽全力贡献。如果有很多测试工作，那么所有的团队成员都可能来帮忙。毫无疑问有些人更擅长测试技术（等等），这并不暗示所有人都是通才，而是指团队成员们共同工作并且互相学习新技术。这样一来，在Sprint计划会议中生成并估计任务时，不必而且也不应该认为所有的任务都由那些能把任务“做得最好”的人来主动承担。而应该是，当需要拿一个新的任务时一次只主动承担一个任务，有目的地兼顾选择那些需要学习的任务（可能通过与这方面的专家结对的方式）。这是为什么不要事先在Sprint计划会议中分配任务的原因之一，而应该在Sprint的过程中“按需”完成。

举个例子来说，很少会出现某一特定任务可能必须要由张三来做，因为这对于其他人来讲需要花太长时间来学习或者根本不可能学会的情况。比方说张三是唯一一个具有某种画图艺术细胞的人，而其他团队成员到死都画不出个简笔小人儿来。如果这种现象并不少见，并且没有随团队的学习而越变越少的話，那一定是有什么地方不对了，可能有必要问一问把所有这些必须由张三来做的画图任务都放在这么短的一个Sprint中是否可行。

很多团队的Sprint待办事项列表采用一个墙面大小的任务板的形式（通常叫做Scrum板），在Sprint过程中，任务（写在报事贴上）在被标为“待办”、“在制品”和“完成”的列之间迁移。参见图5。



图5：可视化管理——贴在墙上的Sprint待办事项列表任务

Scrum的支柱之一是，一旦团队设置了Sprint的目标，任何增加或者改动都只能推迟到下个Sprint。这意味着如果在Sprint的中间产品负责人希望团队进行一个新的事项工作，直到下个Sprint之前他将不能做出这个改动。如果有外部环境情况的出现严重地改变了优先级，并使得团队继续工作下去变成浪费时间，那么产品负责人或者团队可以终止这个Sprint。团队停下来，开始新的Sprint的Sprint计划会议。这么做所带来的破坏通常是有好处的，它抑制了产品负责人和团队诉诸于这种戏剧性变化的动机。

保护团队避免在Sprint过程中改变目标会带来一些强有力的正面影响。首先，团队明白他们的目标绝对不会改变，这加强了团队的关注度，从而保证了工作的完成。其次，它使得产品负责人慎重思考他在产品待办事项列表中排在高优先级并让团队在Sprint中接受的那些事项。

通过遵守这些Scrum的规则，产品负责人得到两样东西。首先，他有信心团队对于尽全力完成他们所选择的一个现实明确的工作集是有承诺的。经过一段时间后，团队对于在更为实际的预期下选择和交付会变得相当熟练。其次，产品负责人要在下一个Sprint开始之前做出他想要在产品待办事项列表中做的改变。在这段时间，增加、删除、修改和重排优先级都是可能和可以接受的。产品负责人却不能改变已经在当前Sprint的开发过程中被选择的事项，他只隔一个或少于一个Sprint的时间就可以实现做出任何修改的愿望。改变方向，改变需求、或者只是改变主意，这些围绕着改变的症状不见了。这可能是为什么产品负责人也和其他人一样热衷于Scrum的原因。

每日Scrum会议

概述：在团队成员间更新信息和进行协调。

参与者：团队必须参加；产品负责人不是必须的；ScrumMaster通常会在场，但要保证团队自己主持会议。

时长：最长15分钟。

一旦Sprint开始，团队就要投身于另一个Scrum实践，那就是每日Scrum会议。在每个工作日指定时间都会举行这个很短的（15分钟，或者更短）会议。团队中的每个人都要参加。为了让它保持简洁，建议每个人都保持站立姿势。这是团队同步他们的工作并互相报告遇到的阻碍的机会。在每日Scrum会议中，每个团队成员一个接一个地向团队的其他成员报告三件事：（1）自上次会议以来完

成了哪些工作？（2）在下次会议前有哪些工作会被做完？（3）遇到了什么阻碍？注意每日Scrum会议并不是向管理者报告状态的会议。这是一个自组织团队互相分享目前工作情况的时刻，从而帮助他们进行工作协调。有人要把这些阻碍记录下来，ScrumMaster负责帮助团队成员解决它们。在每日Scrum会议中只有很少或不会有深入的讨论，其形式只是做回答那三个问题的报告。如果的确需要讨论，那么在每日Scrum会议后马上就会有一个或多个并行的跟进会议，但在Scrum中没要求任何人必须参加这些会议。部分或者全部的团队成员需要针对于他们在每日例会中所收到的信息进行调整时开跟进会议很常见，换言之，这是又一个审视并调整的周期。对于新接触Scrum的团队，通常情况下建议管理者或者其他会被认为有权威的人不要参加每日例会。这样做的风险是会让团队感到“被监视”。它使得团队在压力之下每天都报告出重大进展（这是不切实际的期望），并使团队不愿报告问题。并且它会破坏团队的自我管理，引入微观管理。对于一个利益相关人，更有益的做法是在会议后再找到团队，并帮助解决那些让团队进展慢下来的阻碍。

在Sprint过程中跟踪进度

Scrum中的团队是自管理的，想要做到这一点，团队必须要知道自己做得怎么样。每天，团队成员都会在Sprint待办事项列表（图6）上更新他们对还需要多少工作量来完成他们当前工作的估计。一个也很常见的做法是有人把这些剩余工作量加起来，然后在“Sprint燃尽图”上画点（图7和图8）。这个图显示每天新的对团队完成工作所剩余工作量的估计。理想中，这应该是一条向下的斜线图，其轨迹指向Sprint最后一天“没有剩余工作量”的那一点。所以它被称为“燃尽”图。有时它看上去不错，但通常不会这样。这是产品开发的现实。重要的是它显示了团队向着他们目标的进展，这不是按过去已经花了多少时间（这对于“进展”来讲是个无关的信息），而是按将来还剩余多少工作量来表示的，这也就是团队与他们目标间的距离。如果这条燃尽线在接近Sprint结尾时没有向下接近完成点，那么团队需要做出调整，比如减小工作范围，或者找到在保持可持续步调下更高效的工作方式。

尽管Sprint燃尽图可以用电子表格来创建和显示，很多团队却发现用贴在他们工作空间墙上的纸来展示，用笔来更新效果更好。这种“低技术含量/高感知度”的方案既快捷又简单，并且比电脑上的图表可见性更好。

		每日结束时所剩余工作量的最新估计							
产品待办事项列表事项	Sprint中的任务	志愿者	初始工作量估计	1	2	3	4	5	6
做为买家，我希望把书放到购物车中	修改数据库	王小二	5	4	3	0	0	0	
	创建网页(UI)	赵五	3	3	3	2	0	0	
	创建网页 (Javascript逻辑)	张三和李四	2	2	2	2	1	0	
	写自动化验收测试	尹哲	5	5	5	5	5	0	
	更新买家帮助网页	孙媛	3	3	3	3	3	0	
...									
改进事务处理效率	合并DCP代码并完成分层测试		5	5	5	5	5	5	
	完成pRank的机器顺序		3	3	8	8	8	8	
	把DCP和读入器改为用pRank http API		5	5	5	5	5	5	
...									
总计			50	49	48	44	43	34	

图6：每日在Sprint待办事项列表中更新剩余的工作量

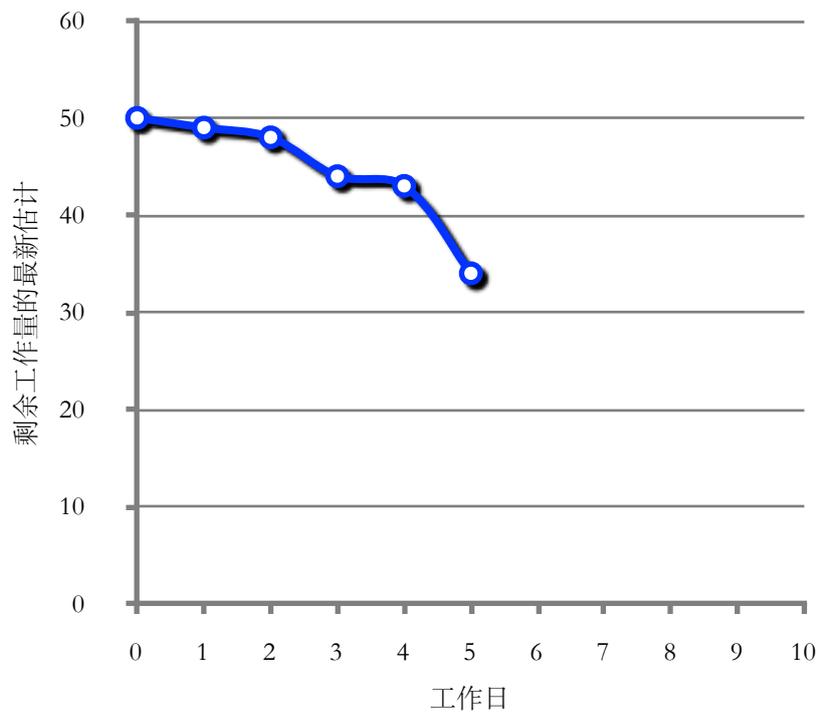


图7: *Sprint*燃尽图

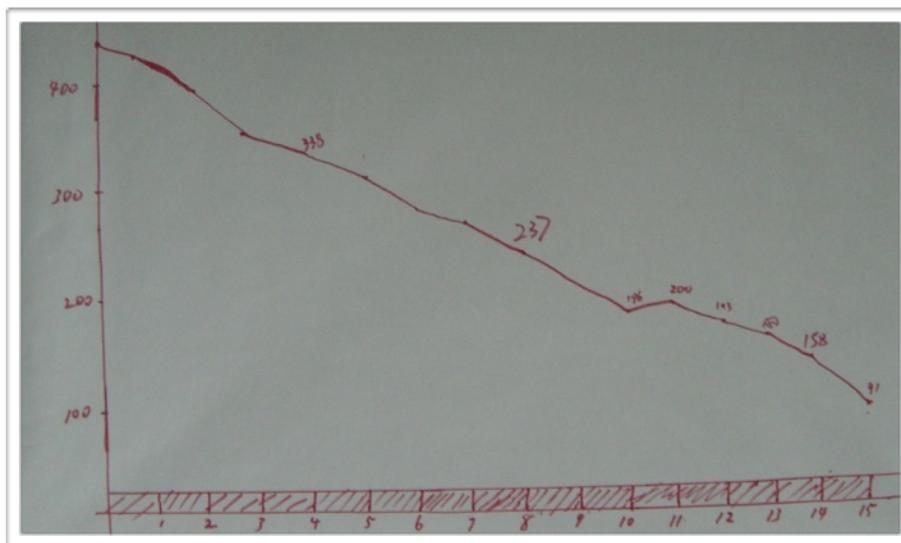


图8: 可视化管理: 手绘*Sprint*燃尽图

产品待办事项列表梳理

概述: 为了将来的*Sprint*拆分大事项、分析事项、重新估计并重排优先级。

参与者: 团队; 如果产品负责人是能够帮助梳理细节的专家, 那么他会全程参与整个活动, 否则他可以只参与其中一部分来设定方向或者重排优先级; 其他理解需求并能给予团队帮助的人; *ScrumMaster*将在会议的开始部分来指导团队如何有效地开这个会, 否则的话他不必参加。

时长: 通常来讲不多于团队一个*Sprint*工作容量的10%, 但有时对于“有大量分析工作”的事项来讲要更长一些。例如, 对于两周的*Sprint*, 可能要有一天的时间花在梳理上。

*Scrum*中一个很少有人知道, 但很有价值的指导原则是, 每个*Sprint*中整个团队要把一定百分比的时间专门用于梳理产品待办事项列表上, 做为对将来*Sprint*的支持。这包含了详细的需求分析、把大的事项拆小、估计新的事项以及重新估计已有事项。*Scrum*中没有说这个工作该如何来完成, 但是

一个经常用到的方法是在接近**Sprint**中部或者结尾时开一个专注的研讨会，这样团队和产品负责人以及其他利益相关人就可以不受打断地专门做这些事情。

这种梳理活动并不是为了当前**Sprint**已经选择的那些事项。它是为了将来的事项，最可能是为了下一两个**Sprint**。有了这个实践方法，**Sprint**计划会议变得相对简单些，因为产品负责人和**Scrum**团队将从一组清晰的，被很好分析过并认真估计过的事项入手。没有开过这种梳理工作研讨会（或者没做好）的特征是**Sprint**计划会议中出现重大的疑问或者发现，或者令人困惑并感觉不完整。这样一来计划工作通常要延续到**Sprint**中去，这显然不是大家想要的。

Sprint评审会议

概述：对于功能性的产品增量进行审视并调整。

参与者：团队、产品负责人、**ScrumMaster**。产品负责人可以邀请其他恰当的利益相关人参加。

时长：时间箱为**Sprint**中的每一周对应一个小时。

在**Sprint**结束后，就到了**Sprint**评审会议，人们会评审这个**Sprint**。出席会议的人有产品负责人、团队成员以及**ScrumMaster**，再加上客户、利益相关人、专家、领导层和任何有兴趣的人。对于两周的**Sprint**来讲最长为两个小时。任何参与者都可以问问题和提意见。

评审会议常常会被错误地打上“演示”的标签，但这并没有抓住这个会议真正的用意。**Scrum**中的一个关键思想是审视并调整。观察并了解发生的情况，然后根据反馈进行演进，不断进行。**Sprint**评审会议是对于产品的审视并调整活动。它让产品负责人了解产品和团队的当前状况（也就是对于这个**Sprint**的评审），也让团队了解产品负责人和市场的当前状况。因此，评审会议中的一个关键元素是团队与产品负责人之间进行深入的对话来了解情况和得到建议等等。评审会议当然要包含使用团队所创造的真的，运行起来的软件，但如果评审的关注点只是在看产品而没有进行交谈，这么做就失衡了。

Sprint评审会议中“运行起来的软件”这部分并不是由团队来做“报告”，不会用到幻灯片。它是指亲手来检验正在运行的真正的软件，例如在一个用于开发的沙盒环境中。在评审会议会议室里会有一台或多台电脑，人们可以在上面检验和使用运行起来的软件。最好有一个积极交互的过程，由真实用户和产品负责人动手与软件交互，而不是由团队做出一个被动的展示过程。

尽量做到在不超过30分钟的时间之内做完**Sprint**评审会议的准备，否则就意味着有什么地方做得不对。

Sprint回顾会议

概述：针对流程和环境进行审视并调整。

参与者：团队、**ScrumMaster**、产品负责人（非必须）。团队可能会邀请其他利益相关人，否则这些人不准参加。

时长：时间箱为对应**Sprint**中的每一周为45分钟。

Sprint评审会议包含了对于产品的审视并调整。而在评审会议之后的“**Sprint**回顾会议”则包含了针对流程和环境的审视并调整。这是团队讨论什么方式能工作，什么方式不工作的机会，并对要尝试的改变达成一致意见。有时**ScrumMaster**可以扮演回顾会议的效率协调者，但可能还是找一个中立的外人来协调这个会议更好。一个好的做法是，**ScrumMaster**们互相帮助协调彼此的回顾会议，这会起到在团队间“交叉受粉”的效果。

组织**Sprint**回顾会议有很多技巧，《敏捷回顾》（《**Agile Retrospectives**》）一书中提供了很有帮助的一系列技巧。

很多团队的回顾会议只关注于问题，这很不好。这使得人们以为回顾会议是某种让人情绪低落的或者负面的事件。正相反，要保证每次回顾会议也关注于正面的事物或者优势。有几本关于“欣赏式探询”的书在这方面给出了更多的点子。

回顾会议如果总是用同样的分析技巧可能就会变得无聊起来，这样的话就要随着时间的推移引入不同的技巧。

开始下一个Sprint

Sprint评审之后，产品负责人可以根据任何新的见解来更新产品待办事项列表，如添加新事项、删除过时事项或修改现有事项。产品负责人负责保证这些变化反映在产品待办事项列表中。图9是个更新了的产品待办事项列表的例子。

优先级	事项	细节 (wiki链接)	初始规模估算	每个Sprint 的新估算					
				1	2	3	4	5	6
1	作为买家，我想把书放入购物车（见wiki页面用户界面草图）	...	5	0	0	0			
2	作为买家，我想从购物车中删除书	...	2	0	0	0			
3	提高交易处理性能（见wiki页面目标性能指标）	...	13	13	0	0			
4	探讨加速信用卡验证的解决方法（见wiki页面目标性能指标）	...	20	20	20	0			
5	将所有服务器升级到Apache 2.2.3	...	13	13	13	13			
6	分析并修复订单处理脚本错误（错误号：14834）	...	3	3	3	3			
7	作为购物者，我想创建并保存愿望表	...	40	40	40	40			
8	作为购物者，我想增加或删除愿望表中的条目	...	20	20	20	20			
...				537	580	570	500		

图9 更新了的产品待办事项列表

Sprint之间没有间歇期，团队通常在某个下午开Sprint回顾会议，第二天早上就直接进入下一个Sprint计划会议（不包括周末休息）。

敏捷开发的一个原则是“可持续的步伐”，团队只有保持合理水平的常规工作时间才可以一直继续这一周期。生产力随着团队实践的演变和铲除影响团队生产力的障碍而增长，而不是通过加班加点或降低质量来取得。

Sprint持续进行直到产品负责人决定产品可以发布。Scrum的完美愿景是在每个Sprint末尾生产出潜在可交付的产品，这就意味着不再需要任何如测试或文档等周边的工作。这意味着每个Sprint中所有的工作都完成了，可以在Sprint评审之后进行交付或发布。然而，许多组织都存在开发实践、工具和基础架构薄弱的问题，并不可能实现这个完美的愿景，因此就需要“发布Sprint”来处理剩余的工作。当需要“发布Sprint”时，我们当这是不得以而为之，这时组织的任务就是要改进他们的实践直到不再需要这个“鸡肋”的做法。

发布管理

有一个时而被提到的问题，即在迭代模式中如何进行长期发布计划。有两种情况需要考虑：1) 第一次发布的新产品，2) 已有产品后来的发布。

在新产品或刚刚采用Scrum的现有产品情况下，在第一个Sprint之前需要进行初始的产品待办事项列表梳理，产品负责人和团队一起塑造一个恰当的Scrum产品待办事项列表。这可能需要几天甚至一周的时间，并且需要召开研讨会（有时称为初始产品待办事项列表创建或发布计划），和一些详细的需求分析，还有被列为第一个发布的所有事项的估算。

在Scrum中出人意料的是，对于已经建立了产品待办事项列表的在建造产品，下一发布应当不需要任何特殊或大量的发布计划。为什么呢？因为产品负责人和团队应该每个Sprint都进行产品待办事项列表的梳理工作（利用每个Sprint百分之五到十的时间），持续不断地为未来做准备。这种“持续的产品开发”模式使得我们不再需要在传统顺序生命周期开发中见到的那种不停打断的准备—执行—结果的阶段性工作方式。

在初始产品待办事项列表梳理研讨会和每个Sprint中持续对待办事项列表梳理的过程中，团队和产品负责人将会进行发布计划，并随着知识的增加而梳理估算、优先级排列和内容。

有些发布是日期驱动的，比如“我们要在11月10日的展销会上发布我们产品的2.0版本”。在这种情况下，团队将会在有限的时间内尽量完成尽可能多的Sprint（而且构建尽可能多的特性）。还有一些产品需要构建特定特性之后才可以被称为完成，产品直到满足了这些需求之后才可以发行（不管要多长时间）。既然Scrum强调每个Sprint生产出潜在可交付代码，那么产品负责人可以选择开始做过渡性发布，以便客户尽快收获成品的好处。

因为他们不可能预测所有的东西，那么重点就在于创建和梳理一份计划以给发布提供一个大的方向，并阐明如何作出折衷决策（比如调整范围还是调整时间表）。把这个当作引导你向最终目的地的路线图。但是究竟选择哪条路线和在行进过程中的决定都可以在途中确定。

“到达目的地比选择何种途径更重要”。

多数的产品负责人会选择一种发布方式。比如，他们会决定一个发布日期，并和团队一起估算在产品待办事项列表中可以在该日期内完成的事项。被加入到当前发布的事项有时称为“发布事项”。在需要“固定价格/固定日期/固定交付内容”承诺的情况下（比如，合同开发），这些参数中的一个或几个就必须有内置缓冲，以顾及不确定性和变化。在这方面，Scrum与其他方法没有不同。

对应用程序或产品的关注

对于应用程序或产品（无论要投入市场，还是在组织内部应用），Scrum让团体远离旧式的以项目为中心的模式，而转向持续进行应用程序或产品开发的模式。从开始、中间到结尾的项目不复存在。因此，也不需要传统的项目经理。而只是一个固定的产品负责人和一个长期、自管理的团队，他们在一系列“无尽的”、固定时长的Sprint中协作，直到产品或应用程序停止开发。所有必需的“项目”管理工作由团队和产品负责人（他可以是内部业务客户或是来自产品管理的人员）共同处理，而不是由IT经理或项目管理办公室人员来处理。

Scrum也可以应用在一次性初创的（而不是创建或演进长生命期的应用程序）真正项目中。但是在这种情况下，团队和产品负责人进行项目管理工作。

如果一个或几个现有应用程序中没有足够的新工作以保证每个应用程序都有一个专注、长期的团队进行工作怎么办？在这种情况下，稳定、长期的团队可以在一个Sprint中负责一个应用程序的事项工作，然后下一个Sprint负责另一个应用程序的事项。这样做使得Sprint的时间通常都非常短（比如一周）。

有时候可能都没有足够的新工作以使用上面的解决方法，那么团队可以在同一Sprint中负责几个不同应用程序事项的工作。但是，要注意这个方法可能会转变成低效的跨越多个应用程序的多任务。在Scrum中的一个基本生产力主题就是在一个Sprint中，团队集中精力于一个产品或一个应用程序的工作。

常见的挑战

Scrum不仅仅是一套具体的实践，确切地说更为重要的是这个框架提供了透明度并给出了“审视并调整”的机制。Scrum通过让影响产品负责人和团队效率的机能失调和障碍变得显而易见，使它们得到及时处理。比如，产品负责人可能不清楚市场情况、特性或如何估算它们的相对商业价值。又或者团队可能对于工作量估算或开发工作还不够熟练等。

Scrum框架会迅速揭露这些弱点。Scrum并不能解决开发的问题，只是让问题都赤裸裸地显现出来，并为人们尝试用短周期和小改进试验来解决问题提供了框架。

假设团队因为任务分析和估算技能不佳，而不能交付他们预期在首个Sprint完成的工作。对于团队而言，这是次失败。但是实际上，这是个必需经历的第一步，以使得团队对于以后的预期更加实际和周到。Scrum的这种模式使得机能失调显现出来，让团队能够及时处理。正是这种基本的机制带给了使用Scrum的团队所能经历到的最大好处。

一个常见的错误做法是当使用某个Scrum实践遇到挑战时改变Scrum。比如，团队交付有困难，他们可能决定延长Sprint，那么时间永远都是充裕的，并且在这过程中确保自己永远都不用学习如何更好地估算和管理时间。这样，没有经验丰富的ScrumMaster的教练和支持，这个组织的Scrum就会

蜕变成只是其自身弱点和功能失调的镜像，并且破坏了Scrum所能带来的真正利益：也就是使得好坏都显露无遗，给组织提供自我提升的机会。

另一个常见的错误做法就是假定某个实践是不准许或禁止的，只是因为Scrum没有明确地提到。比如，Scrum没要求产品负责人为产品制定长期策略，也没要求工程师对于复杂的技术问题向经验丰富的前辈讨教。Scrum将这些都留给相关的人来做正确的决定，在多数情况下，以上两个实践（和其他一些）都是特别推荐的。

还有一个需要注意的是管理者强制团队使用Scrum。Scrum提供给团队空间和工具来自我管理，这种从管理层强压下来的命令并不是取得成功的方法。一个比较好的做法就是让团队先从同事或管理者处了解到Scrum，进行全面的专业培训，然后由团队决定在一定的时期内忠实遵循Scrum实践，在该时期结束时团队将评价其工作经历，再决定是否继续使用Scrum。

虽然首个Sprint通常对于团队来说是极富挑战性的，但值得庆幸的是在Sprint结尾时Scrum带来的好处就会显现出来。所以很多新的Scrum团队都惊叫：“Scrum太难了，但是比起我们以前的做法它简直是太好了！”

附录A：推荐阅读

关于Scrum已经有很多发表了的材料。在这个参考部分，我们想推荐一些线上材料和几本书以供阅读。

线上材料：

The Lean Primer - An introduction to Lean Thinking, an important influence to Scrum.
<http://www.leanprimer.com>

The Distributed Scrum Primer - Additional tips for teams who aren't co-located.
<http://www.goodagile.com/distributedscrumprimer/>

The ScrumMaster Checklist - A list of question that good ScrumMasters use.
<http://www.scrummasterchecklist.org/>

Feature Team Primer - Scaling Scrum with Feature Teams,
<http://www.featureteams.org>

The Agile Atlas - Core Scrum. ScrumAlliance description of Scrum.
<http://agileatlas.org/atlas/scrum>

Scrum Guide - Scrum.org description of Scrum.
<http://www.scrum.org/Scrum-Guides>

Agile Contracts Primer - How to make Scrum-friendly contracts.
<http://www.agilecontracts.org/>

书籍：

《Leading Teams》 作者： Richard Hackman

《精益和敏捷开发大型应用指南》 作者： Craig Larman, Bas Vodde

《精益和敏捷开发大型应用实战》 作者： Craig Larman, Bas Vodde

《Scrum敏捷项目管理》 作者： Ken Schwaber

《Scrum敏捷软件开发》 作者： Mike Cohn

附录B：术语

燃尽

在一个Sprint中、一个发布中或者一个产品中，剩余工作量随时间变化的趋势。原始数据来自Sprint待办事项列表和产品待办事项列表，剩余工作量记录在纵坐标上，时间（例如Sprint中的天数，Sprint个数等）记录在横坐标上。

每日Scrum会议

每个团队每天召开的短会，在这过程中团队成员检验他们的工作，对工作和进度进行同步并把阻碍报告给ScrumMaster来移除。在每日Scrum会议之后可能会开跟进会议来调整后面的工作以优化这个Sprint。

开发团队

团队的另一个名称。

完成

所有参与者都认为是完成，并遵循了组织的标准、规范和指导原则。当在Sprint评审会议中把某些东西描述为“完成”时，它必须符合这个达成一致的定义。

剩余工作量估计（Sprint待办事项列表中的条目）

一个团队成员仍需要工作在某任务上的小时数的估计。当工作在这项Sprint待办事项列表中的任务时，它的这项估计在每天结束时要进行更新。估计的值为所有预计剩余的工作量，这和多少人工作无关。

增量

产品功能，由团队在每个Sprint中开发，是潜在可交付的，或者对于产品负责人的利益相关人来讲是有用的。

潜在可交付产品功能增量

整个产品或系统的一个完整的一部分，如果产品负责人或利益相关人选择实现它，那么它将是有益的。

Sprint

一次迭代，或者一个相似工作的重复周期，为产品或系统产出增量。不会超出一个月，通常大于一周。全部工作过程中的Sprint时长是固定的，工作在同一系统或产品上的团队使用同样长度的周期。

产品待办事项列表（Product Backlog）

一个排好优先级的需求列表，包含了把它们变成完成的产品功能所需的时间估计。在产品待办事项列表中优先级越高的事项的估计就越精确。随着业务条件和技术的变化，这个列表的内容会不断涌现和改变。

产品待办事项列表事项

功能性需求、非功能性需求和问题，按照对于业务的重要性和依赖关系排优先级，并且对它有估计。估计的精确程度取决于产品待办事项列表事项的优先级和粒度，在下个Sprint可能会被选中的最高优先级的事项粒度很小并且很精确。

产品负责人

负责管理产品待办事项列表的人，也是负责最大化产品价值的人。产品负责人负责代表所有与项目及其所产生产品有利害关系人的利益。

Scrum

这不是个缩写，是在英式橄榄球中死球后再度开球的机制。

ScrumMaster

负责Scrum流程、对它的正确应用以及把它的好处最大化的人。

Sprint待办事项列表

团队在Sprint中工作的列表。它通常被分解为一组更详细的任务。这个列表在Sprint计划会议中出现，可能会由团队在Sprint中更新，按需删除条目或增加新任务。Sprint待办事项列表中的每个任务在Sprint中都会被跟踪记录，并显示出其剩余工作量估计。

Sprint待办事项列表中的任务

团队或团队中的某成员为了把承诺的产品待办事项列表条目变成系统的功能而定义的一个任务。

Sprint计划会议

一个时间箱为4小时的会议（对于两周的Sprint来讲），开启每个Sprint的序幕。会议被分为两个2小时的片断，每个也受时间箱控制。在第一部分，产品负责人把产品待办事项列表中的高优先级部分展示给团队。团队和产品负责人一起合作来帮助团队决定他们能把多少产品待办事项列表在下一个Sprint中变成实际的功能。在第二部分，团队通过做设计和把工作分解来计划他们将如何做到这些，这样他们就知道如何来达成Sprint目标。

Sprint回顾会议

一个由ScrumMaster协调的会议，整个团队讨论刚刚完成的Sprint并找出哪些改变可能会使得下一个Sprint更有趣或更高效。

Sprint评审会议

在每个Sprint的最后一个时间箱为两小时的会议（对于两周的Sprint），团队与产品负责人还有利益相关人合作，一起检验这个Sprint的产出。这通常开始于对已经完成的产品待办事项列表事项进行评审，然后是关于机遇、约束和风险的讨论，再然后讨论下一步最好做什么（可能会导致产品待办事项列表的改变）。只有完成的产品功能才可以用来演示。

利益相关人

与项目的产出有利益关系的人，可能因为他们是出资的人、使用者或者是会被它影响到的人。

团队

一组跨职能工作的人，负责自我管理来在每个Sprint开发产品增量。

时间箱

事件或会议的一个不允许超过的时间段。例如，每日Scrum会议的时间箱为15分钟，无论如何都会在15分钟后终止。对于会议，实际长度可以短一些。对于Sprint，长度是不变化的。