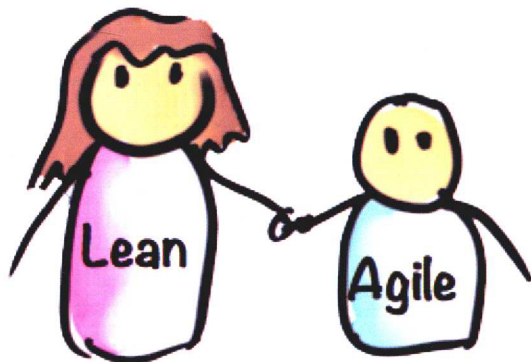
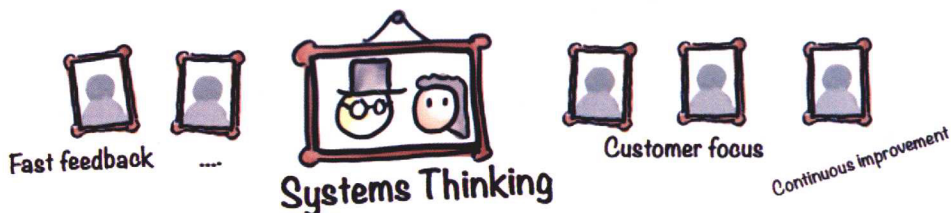


# Lean & Agile are siblings



## 走出硝烟的精益敏捷 我们如何实施Scrum和Kanban

[瑞典]亨利克·克里伯格 马蒂斯·斯加林 著  
李剑 译

杰夫·苏瑟兰 | 迈克·科恩 | 玛丽·波彭迪克 | 大卫·安德森 | 吴永强 | 金毅

—— 推荐序 ——

- 👤 真实回放全球知名实干家的精益敏捷实施过程
- 👤 结合实例回放，充分演绎敏捷方法的选用智慧
- 👤 引领国内近万名读者进阶敏捷精益的启蒙读物
- 👤 中小型团队（项目）不可或缺的敏捷精益入门

清华大学出版社

# 走出硝烟的精益敏捷

## 我们如何实施Scrum和Kanban

[瑞典]亨利克·克里伯格 马蒂斯·斯加林 著  
李剑 译

清华大学出版社  
北京

## 内 容 简 介

本书真实反映了一个团队的精益敏捷落地过程。第 I 部分介绍了团队是如何实施主流敏捷方法 Scrum 的。主题涵盖如何写产品列表, 如何准备、制定、公开和编写计划, 如何布置团队空间, 如何开每日站会, 如何做演示和回顾, 如何对待固定价格的合同, 如何结合使用 Scrum 和 XP, 如何做测试, 如何管理多个团队, 如何管理分布式团队。最后, 作者还给出了一个很有价值的 ScrumMaster 检查清单。第 II 部分主要介绍 Scrum 和 Kanban 的结合使用。在对比两者之后, 作者通过一个具体的案例来说明如何搭配使用两种方法来实现价值最大化。

本书行文风趣, 具备较强的知识性和可读性, 适合所有打算导入并实施精益敏捷的软件从业人员阅读和参考。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

### 图书在版编目(CIP)数据

走出硝烟的精益敏捷: 我们如何实施 Scrum 和 Kanban/ (瑞典) 亨里克·克里伯格, (瑞典) 马蒂斯·斯加林著; 李剑译. —北京: 清华大学出版社, 2019.10

ISBN 978-7-302-53863-9

I. ①走… II. ①亨… ②马… ③李… III. ①软件开发—项目管理 IV. ①TP311.52

中国版本图书馆 CIP 数据核字 (2019) 第 207185 号

责任编辑: 文开琪

封面设计: 李 坤

版式设计: 方加青

责任校对: 周剑云

责任印制: 杨 艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者: 涿州汇美亿浓印刷有限公司

经 销: 全国新华书店

开 本: 178mm×230mm 印 张: 12.5 字 数: 275 千字

版 次: 2019 年 11 月第 1 版 印 次: 2019 年 11 月第 1 次印刷

定 价: 69.80 元

产品编号: 083107-01

# 推荐序1

杰夫·苏瑟兰（Jeff Sutherland）  
Scrum联合创始人，《敏捷革命》作者

开发团队需要了解一些Scrum的基础知识。该怎样创建产品backlog，对它进行估算？怎样把它转化成sprint backlog？怎样管理燃尽图（burndown chart），计算团队的生产率（velocity）？Henrik的书可以用作一些基础实践的入门指南，帮助团队从试用Scrum中成长，最终成功地实施Scrum。

当前，良好的Scrum执行过程对需要风险投资的团队正在变得日益重要。我现在是一个风险投资团队的敏捷教练。为了帮助他们达成目标，我给出一个建议：“只投敏捷实践实施情况良好的敏捷公司。”团队中的高级合伙人（senior partner）向所有待投资的企业提出同一个问题：“你们是否清楚团队的生产率？”目前他们都很难做出明确的答复。要想在将来得到投资，开发团队必须清楚自己的软件生产率。

为什么这一点如此重要呢？如果团队不清楚自己的生产率，那么产品负责人（product owner）就无法用可靠的发布日期来创建产品路线图。如果没有可靠的发布日期，公司的产品就可能失败，投资人的钱就有可能化为乌有！

无论公司规模大小，创办时间长短，或者是否有资金注入，这个问题都是必须要面对的。最近在伦敦举办的一个大会上，我们讨论谷歌内部的Scrum实施状况，当时的听众有135人，我问他们中有多少人在用Scrum，只有30个人举手。我接着又问他们是否在根据诺基亚标准来做迭代开发。迭代开发是敏捷宣言的基本原则——在早期频繁地交付可工作的软件。诺基亚用几年时间对上百个Scrum团队的工作进行回顾，总结出迭代开发的基本需求。

- 迭代要有固定时长（称为“时间盒”，即timebox），不能超过6个星期。
- 在每一次迭代的结尾，代码都必须经过QA的测试，能够正常工作。

使用Scrum的30个人里面，只有一半人说他们遵守了诺基亚标准，符合敏捷宣言的首要原则。我又问他们是否遵守了诺基亚的Scrum标准。

- Scrum团队必须要有产品负责人，而且团队都清楚这个人是谁。
- 产品负责人必须要有产品backlog，其中包括团队对它进行的估算。

- 团队必须要有燃尽图，而且要了解他们自己的生产率。
- 在一个sprint中，外人不能干涉团队的工作。

仅有的30个实践Scrum的在场人士中，只有3个能够通过诺基亚的Scrum测试。看来只有这几个团队才有可能在将来得到我这些风险投资伙伴的钱了。

如果按照亨里克列出的实践执行，你会拥有如下产物：产品backlog、对于这个backlog的估算、燃尽图；你会了解团队的生产率，并掌握切实有效的Scrum过程中所包含的众多基础实践。这些收获就是本书的价值所在。你将通过诺基亚的Scrum测试，对工作的投资也会创造价值。如果你的公司还正处于创业阶段，也许还会收到风险投资团队的资金注入。你也许有望参考打造软件开发的未来，成为下一代软件产品中领军产品的创建者。

## 推荐序2

迈克·科恩 (Mike Cohn)

《敏捷软件开发：用户故事实战》和《敏捷软件开发：Scrum成功之道》作者

Scrum和极限编程 (XP) 都要求团队在每一次迭代的结尾完成一些可以交付的部分工作成果。迭代要短，有时间限制。将注意力集中于在短时间内交付可工作的代码，这就意味着Scrum和XP团队没有时间进行理论研究。他们不会花时间用建模工具来画UML图和编写完美的需求文档，也不会为了应对在可预计的未来中所有可能发生的变化而去写代码。实际上，Scrum和XP都关注如何把事情做好。这些团队承认在开发过程中会犯错，但是他们明白：“要投入实践中，动手去构建产品，这才是找出错误的最好方式；不要只是停留在理论层次上对软件进行分析和设计”。

注重实践而非理论研究，这正是本书的独到之处。亨里克·克里博格 (Henrik Kniberg) 很清楚，初涉门径的人更需要这种书籍。他没有对“什么是Scrum”进行冗长的描述，只是给出了一些网站作为参考。从一开始他就在讲他的团队如何管理产品backlog，并基于它进行工作。接着他又讲述了成功的敏捷项目中包含的所有元素和实践。没有理论，没有引用，没有脚注，没有废话。亨里克的书没有从哲学角度上分析为什么Scrum可以工作，没有分析为什么你可能会尝试不同的选择。它描述的是一个成功敏捷团队的工作过程。

所以本书书名中的“我们如何实施Scrum”才显得格外贴切。这也许不是你实施Scrum的方式，这是亨里克的团队实施Scrum的方式。你也许会问：“为什么我要关心别的团队怎样实施Scrum？”这是因为通过关注其他团队的实施过程，尤其是成功的案例，我们可以学到更好的实施方式。这不是，也永远不会是“Scrum最佳实践”的罗列，因为团队和项目的真实场景比其他一切重要得多。我们应该了解的是优秀实践及其应用范围，而不是最佳实践。在读过足够多的成功团队的实践经验以后，你便会做好充分的准备，从容面对实施Scrum和XP的过程中会遇到的艰难险阻。

亨里克提供了很多优秀实践，还有对应的使用场景。通过它们，我们能够更好地掌握如何在自己的项目中，在充满硝烟的战壕里使用Scrum和XP。

## 推荐序3

---

玛丽·波朋迪克（Mary Poppendieck）

亨里克·克里伯格（Henrik Kniberg）身上有一种稀有的特质，他可以从复杂的问题中剥离掉无关的因素，提取出最核心的思想，以最清澈、最透明的方式讲述出来，让人觉得这些知识竟然前所未有地简单易懂。他在这本书里完成了一项壮举。他解释了Scrum和Kanban的区别，让读者明白它们不过是工具而已，我们所要做的是打造一个完备的工具箱，彻底弄懂每一项工具的长处和局限，明白什么情况下用什么工具。

在这本书中，你会了解到Kanban所要解决的问题域、它的强项和弱点、使用时机。你也会学到很精彩的一课——Scrum或是你手头任何一款工具的改进时机和方案。亨里克在书中讲得很明白，一开始选用什么工具并不重要，重要的是你要不断改进它的用法，不断扩充自己的工具箱。

马蒂斯·斯加林（Mattias Skarin）讲述了Scrum和Kanban在真实场景中的应用，把这本书变得更加精彩纷呈。这两个工具或单兵突击，或联合作战，都在推动着软件开发过程改进的步伐。他们讲到，世界上没有“最好”的方式，每个人都需要在自己的场景中思考，找到继续前行的方向。

## 推荐序4

大卫·安德森 (David Anderson)

Kanban的本质是一个很朴素的思想：在制品 (work-in-progress, WIP) 必须要限制。只有当前的某项工作完成交付或是有了来自下游的拉动，新的工作才能开始。Kanban (或信号卡) 的含义是，因为当前的工作没到限额，有新任务可以拉进来，于是发出一个肉眼可见的信号。这件事情听上去并不是革命性的变化，也似乎不会对团队和组织的绩效、文化、能力及成熟度产生多么深刻的影响。但它却奇迹般地做到了！Kanban看似不打眼，却能影响业务的方方面面。

我们意识到，Kanban是一种改变管理方针的途径。它不是软件开发、项目管理的生命周期或者流程。它是给现有的软件开发生命周期或者项目管理方法中引入变化的途径。实施Kanban的原则是把当前的工作作为起点，通过价值流分析来理解当前的流程，然后为每个环节中的在制品上限达成共识，让Kanban信号拉动着工作流动起来。

Kanban在敏捷软件开发的团队中成效显著，但也吸引了采用传统开发方式的团队的目光。在对组织文化进行精益改造，推动持续改善的过程中，Kanban往往会被先行引入。

因为WIP在Kanban中是受限制的，所以不管是什么任务因为什么原因受阻，都会对整个系统造成严重的影响。受阻的任务到了一定数量，就会导致整个流程没法运作。无论是团队还是组织，都要聚焦于解决问题并让任务重新流动起来。

Kanban使用可视化管理的方式，跟踪任务在整个价值流中流经的不同阶段。人们通常会用带贴纸的白板或是电子卡片墙。我觉得最好的方式是两个都用。它带来的透明化也推动着文化上的改变。敏捷方法在某些方面已经有了很好的透明管理，例如在制品、已完成的工作、度量 (如生产率——在一个迭代中完成的工作数量)；但Kanban做得更深入，它让流程中的所有环节以及工作的流动状态完全呈现在我们眼前。它让人们看到瓶颈、队列、变化和浪费。如此种种都会影响我们交付的有价值的成品数量，影响循环周期，从而影响组织的效益。团队成员和外部相关干系人都可以通过Kanban看到自己的行为 (和无为) 带来的影响。一些早期的案例分析表明，Kanban改变了人们的行为方式，促进了工作中更为紧密的协作。当人们看到瓶颈、浪费和变化所造成的影响之后，就会开始讨论如何做出改善，团队也会很快把改善方案落实到过程中去。

由此可见，Kanban所提倡的是渐进式演化，逐渐向敏捷和精益的价值观靠拢。它并不要求像秋风扫落叶一般，把过去的工作方式一扫而光；它提倡的是“随风潜入夜，润物细无

声”。它的改变是所有人彻底理解并达成共识的。

Kanban具有拉动系统的本质，提倡推迟承诺——不管是对新任务排序，还是交付当前手头的工作。一般来讲，团队会跟上游的相关干系人定期开会，排定接下来要做的工作。这种会议因为时间较短，所以会经常开。干系人在会上要回答一个很简单的问题，例如，“到现在我们又有了两个空出来的位置。我们的循环周期是6周，你最希望在6周之后交付哪两项任务？”这个问题有双重含义。首先，一个简单的问题可以得到快速而明确的答复，缩短会议时间；其次，它还意味着我们要到了最后责任时刻才开始承诺接下来做什么事情。这可以更好地管理用户期望，缩短从承诺到交付的循环周期，而由于优先级变化的可能性减少，返工的几率也就少了。

我要说的最后一点是，制定WIP的限额可以让循环周期更容易预测，让交付更加可靠。出现阻碍或者bug之后的“停线”机制，可以确保质量水准，让返工情况急剧降低。

虽然书中的通透解释隐约映照出一些成长印记，但我们已经没办法还原出Kanban完整的发展路径。Kanban不是在一个短短的午后时光里靠着天赐般的灵感喷涌孕育出来的。它用了多年时间才渐渐形成。那些在文化、能力、成熟度和组织等各方面带来奇迹般改变的哲学和社会学因素都是我们未曾想象过的，但它们切切实实地发生在了眼前。Kanban引发的很多结果都是反直觉的。看上去机械化的操作——WIP上限和拉动式生产——却会对人与人之间交互、协作的方式产生生长久深远的影响。不管是我，还是其他人，在刚开始使用Kanban的时候，都没有预想到这一点。

我曾经努力寻找一种阻力最小的改变手段，这才有了后来的Kanban方法。那是2003年发生的事情。一开始我追求的是机械效益，后来在实施精益技术的时候发现，既然管理WIP是有意义的，那么限制它就更有意义了，这还能解放一部分管理方面的精力。于是从2004年起，我就决定从几个首要原则开始推行拉动式系统。微软的一名经理找到我，问我能不能给他的团队引入一些改进，他们当时正在做内部IT系统的维护升级。我最开始用的是基于约束理论的拉动生产方案Drum-Buffer-Rope，效果非常好，循环周期缩短了97%，产出高了三倍，可预测性超出了98%。

2005年，唐纳德·雷勒特森（Donald Reinertsen）劝我把Kanban全面落地。2006年，我得到一个机会，掌管西雅图Corbis公司的软件工程部。2007年，我开始展示实施Kanban的成果。最开始，我在2007年芝加哥的精益新产品开发峰会上做了演讲。同年8月，又在Agile 2007大会的开放空间会议上进行了分享。当时有25人参加，有三个来自雅虎，他们分别是亚伦·桑德斯（Aaron Sanders）、卡尔·斯哥特兰（Karl Scotland）和乔伊·阿诺德（Joe Arnold）。他们各自回到美国加利福尼亚、印度和英国以后，就开始在挣扎着实施Scrum的团队中推行了Kanban。他们还创建了一个雅虎讨论组，在我写下这篇文章的时候，这个

讨论组已经有了800个会员。Kanban正在广泛传播中，先行者开始了经验分享。

现在是2009年，Kanban的实施越来越普遍，我们看到了越来越多的一线报告。过去的5年里，我们从Kanban上面学到了很多，也会持续学习下去。我现在的工作重心就是实施Kanban、记录Kanban、讲述Kanban和思考Kanban，一切都是为了更好地理解Kanban，把它更好地传播给其他人。我渐渐地不再讨论Kanban和其他敏捷方法的区别，只在2008年的时候花了些时间解释为什么Kanban可以跟敏捷兼容。

像“Kanban跟Scrum比起来怎么样？”这种问题，还是留给更有经验的人来回答吧。我很高兴亨里克和马蒂斯成为了这方面的领军人物。作为知识工作者的你，需要信息来辅助决策和指明道路。亨里克和玛蒂斯以我所不能及的方式满足了你的需求。我尤其欣赏亨里克幽默的文风和他尊重事实且不固执己见的做事方式。他笔下的卡通像和图片远远胜过长篇累牍的文字。马蒂斯的案例分析也很重要，它用事实证明了Kanban不仅仅是理论而已，也许对你的组织会起作用。

我希望你喜欢这本书。它会让你更加深刻地理解敏捷，尤其是Kanban和Scrum。如果想了解更多Kanban的知识，请访问我们的社区网站：The Limited WIP Society, <http://www.limitedwipsociety.org/>。

## 推荐序5

---

吴永强

斑马资本合伙人，去哪儿网前CTO

《敏捷宣言》发表于2001年，那时正值第一代互联网公司的巅峰时期。自那以后，敏捷开发被不断实践、改进和完善，到今天已经被全球各行各业熟悉和接受。我可以毫不夸张地说，敏捷已经成为最主流的软件开发模式，它产生了广泛而深远的影响。

对于初次接触的读者，如果把敏捷开发放到一个宏观的背景下来观察，就更能清晰地理解敏捷开发思想的发展源头和脉络。

以雅虎为代表的第一代商业互联网公司在20世纪90年代的巨大成功标志着人类进入了互联网时代，后续的以搜索和电商为代表的第二代互联网公司，以移动互联网和社交网络为代表的第三代互联网公司，和当前以大数据、AI、IOT为特征的第四代互联网公司正在席卷世界。以信息技术为核心的第四次工业革命不仅对各行各业带来深远的影响，对软件开发的模式也有重要的推动作用。

首先，信息技术的普及和大规模运用带来商业竞争环境的剧烈变化，ERP、MIS、CRM、在线交易和支付在行业中被广泛采用，同时消费者信息获取手段的演进，特别是搜索、推荐、移动互动互联网的出现，使得个人和组织能够全天候、跨地域、大容量地获取和处理各种信息，使原先依赖信息不对称、不透明的商业模式越来越难以为继，用户/客户需求变化越来越快，越来越个性化。商业竞争日趋激烈，原先以规划、预测为基础的商业模式被迫演变成更具创造性的、以探索、验证、反馈、修正和改进为基础的小步快跑的商业模式，从这个角度看，软件开发领域的敏捷革命有着其内在的商业驱动力。

其次，以IP网络为基础的互联网、移动互联网和IOT万物互联网，共同构筑了一个参与设备繁多，随时随地在线的基础平台，人类第一次有能力以极低成本部署和更新大规模IT软件系统。同时，软件/系统的架构也发生巨大的变化，SOA、大规模中间件和API化等分布式架构的建立和采用，系统耦合程度的降低，让我们可以在一个更稳定、更小型化的系统环境里工作。部署成本的大幅降低和分布式系统架构的演进，使敏捷开发所主张的以业务需求为核心、快速迭代交付成为可能，并产生了巨大的价值。

最后，知识经济的最大的特征就是人成为最重要的生产要素，人力资源变成企业的核心战略资源，如何发挥个人的创造性、提高人的生产效率，是企业寻求战略竞争优势的重要途径，而敏捷开发正好前瞻性地解决了这个问题。创造而非执行，赋能而非命令，敏捷开发

中强调的小团队、信息透明、沟通密集，都有助于知识工作者在充分发挥自身的创造力的同时凝聚成关系紧密、自组织、具有超高生产力的团队。

本书作者用短短的篇幅生动描述了Scrum敏捷开发模式实施中的核心步骤，书中并没有关于敏捷开发的理论阐述，而是通过详细记录作者实施Scrum敏捷开发的日常实践、实践过程中遇到的冲突和问题以及相应的解决方法，使得读者可以按图索骥，正确实施Scrum敏捷开发，并解决过程中遇到的相关问题。书中相关的图表极具实用性，可以直接拿来运用到实践中。有心的读者也能从书中内容澄清一些常见和重要的误区：譬如只强调成员自驱动而放弃了日常的管理。实际上敏捷开发管理中采用的每日例会、任务板、燃尽图等手段，比传统的开发管理方式更加动态和细致，这也是敏捷开发有更高生产力同时能应对高度动态的业务要求的重要原因。再如忽视计划的重要性，其实作者在书中反复强调Sprint计划会议的重要性，只是相比于传统的开发计划，敏捷开发的计划更强调以业务需求为中心、团队成员取得共识、可衡量的交付结果以及固定的时间窗口等。

信息论之父香农定义“信息是用来消除随机不确定性的东西”，所以进入到信息时代之后，竞争的核心就是谁能更好地收集、处理信息，更快地根据信息做出更好的决策，并能更快收集针对决策的反馈。对国家如是，对企业和个人亦如是。而“敏捷”则是应对上述挑战的一个重要的解决思路，同时也是一个重要的实践方法。读者若能将敏捷的思维用于日常的学习、职业和创业生活中，或许可以得到很多意想不到的启发和效果。

## 推荐序6

---

金毅，敏捷咨询师

承蒙文老师的厚爱，让我获得了这个学习机会，并再次思考和总结。

亨里克是我非常钦佩的一位瑞典精益敏捷顾问，他在Crisp咨询公司的BLOG上这样描述自己：“I debug, refactor, and optimize IT companies.（我调试，重构，并优化IT公司）。”

亨里克（Henrik Kniberg）以及后来与马蒂斯合著的两部作品脍炙人口，可以为不同层次的敏捷实践者带来诸多收益。作为一位高产作者，亨里克的文章、书还有各种演讲总是用案例、细节和数据说话，总是能够使同样是敏捷顾问的我感到信服和共鸣。

此外，他还不断尝试推陈出新，与客户共创了很多成功案例或者成功模式，比如他在2012年发布的Spotify规模化敏捷模式，就是他与客户长期合作共同打磨出来的成功的敏捷组织的运作模式。我在这两年为国内银行组织落地实施敏捷部落制时，还会时常参考他的文章。还有就是2016年他发布的乐高敏捷转型案例，只花1天时间就完成了150人的PI计划会，这些成就都是敏捷圈子里的佳话和里程碑。

任何规模化敏捷的基础仍然是小团队的敏捷，而小团队敏捷就离不开看板和Scrum这两种最常用的方法。今年，我为一家银行客户快速启动敏捷团队试点时，就是一天之内建立了物理看板，使用定制的需求卡片和任务便利贴可视化当前项目状态，并召开了第一次站会，这样就开启了一个小团队的敏捷之旅。继而演练了需求梳理和拆分，迭代回顾会等。让一个团队快速感知敏捷是什么？快速带来好处，也同时暴露主要问题。

尽管没有什么敏捷方法是银弹，但是因为存在这些成熟而简单易行的方法，我们至少多了一些抓手来与团队进行交互，更快地了解团队，探测问题，并持续演进和调整。总之，Scrum和看板方法值得敏捷初学者入门时学习，更值得敏捷实践者深入实践，结合实际情况总结出自己的打法，这是每个人都绕不开的学习之路。

亨里克的书提供了精辟的理解，让你在学习和实践中很容易产生共鸣，激发你的思考，鼓励你去尝试。最重要的是，这本书异常精炼，你可以把它当成一个随身携带的小工具箱，为自己的敏捷学习和实践提供实时帮助。

精益敏捷的从入门到精通，很多人都是从这本书开始的，期待你早日拿起这本书，愉快地开启你的精益敏捷旅程！

# 译者序：敏捷不是说出来的，是……

---

李剑

孙子兵法有云：“兵无常势，水无常形，能因敌变化而取胜者谓之神。”很多人都向往用兵如神的境界，想必也知道“读万卷书不如行万里路”，纸上谈兵的故事更是耳熟能详，但偏偏不能举一反三。

且看风清扬的一段话：“……你将这华山派的三四十招融会贯通，设想如何一气呵成，然后全部将它忘了，忘得干干净净，一招也不可留在心中。待会便以什么招数也没有的华山剑法，去跟田伯光对打。”如果有人问，既然“无招胜有招”是武学的最高境界，那干脆什么招数都不要学，拿把剑乱挥乱舞，处处破绽，也就处处无破绽，便是天下第一了。听到这话的人肯定会笑他“太缺心眼”。

我在这里不想解释为什么上面那种说法缺心眼，因为不缺心眼的读者肯定能够理解说他缺心眼的理由。但有句话叫“不识庐山真面目，只缘身在此山中。”对待离自身尚远的事物时，人们可以把它分析得淋漓尽致；但到了自己身上，却往往陷入“当局者迷，旁观者清”的境界，譬如青春，譬如爱情，譬如敏捷软件开发。

我想，这本书的读者大概都知道，现如今敏捷开发是何等炙手可热，但潮流一起，跟风者势必为之。虽然没法在这篇短短的序中逐一批驳，但大家也可以仔细思索一下，在周边是否存在缺心眼的做法。比如，把各种bad smell背下来以后就大谈特谈重构的好处；版本控制、缺陷跟踪和配置管理等一无所知，便好高骛远，一味追求持续集成；单元测试还不会写，就疯狂宣传测试驱动开发……这些都还好，更有甚者，把敏捷等同于迭代，等同于又敏又捷，又快又爽；这也无所谓，只要没有在实际上对敏捷一无所知，对想要达到的目标不甚了了，对项目中存在的问题视若无睹的情况下宣传敏捷，推行敏捷就可以了。但如果前面那些条件都吻合，最后这一点还能不是水到渠成的事么？

其实，敏捷不是说出来的，是干出来的。

是为序。

# 嘿，Scrum成了！

Scrum成了！至少对我们来说它已经成功了（这里指的是我当前在斯德哥尔摩的客户，名字略过不提）。希望它对你们也一样有用！也许这本书会对你们实施Scrum的过程有所助益。

这是我第一次看到一种开发方法论（哦，对不起，Ken，它是一种框架）可以脱离书本成功运作。它拿来就能用。所有人——包括开发人员、测试人员和经理——都为此而高兴。它帮助我们走出了艰难的境地，而且让我们在剧烈的市场动荡和大规模的公司裁员中依然能够集中精力在项目上。

我不该说我为此感到惊讶，但实情确实如此。在一开始我大致翻了几本讲Scrum的书，它们把Scrum描述得挺不错，却给我留下了一种太过美好以致于不太真实的感觉（我们都知道“某些东西看上去太好了……”这类说法的含义）。所以我没法不对它有一丁点儿怀疑。但在使用Scrum一年以后，先前的零星疑虑早已烟消云散。我被它深深地震撼了（我们团队中的大部分人都和我一样），以后只要没有充分的理由来阻止我，我都会继续使用Scrum。

# 致谢

本书初稿完成仅用了个周末，但很显然，那是一个超高强度工作的周末！投入程度高达150%。

感谢我的妻子索菲亚（Sophia）和两个孩子大卫（Dave）与詹尼（Jenny），我那个周末扔下他们独自工作，他们对此表示了宽容；还要感谢我的岳父母伊娃（Eva）和尤根（Jörgen），在我忙碌的时候，他们过来一起照看整个家庭。

同时，还应该感谢在斯德哥尔摩Crisp工作的同事，还有Scrum Development Yahoo 讨论组的成员，他们一起校稿，提出了很多改进意见。

最后，我要深深感谢所有的读者，从你们长期的反馈中我收获颇丰。尤其要指出一点，能够通过本书点燃许多人尝试敏捷软件开发的热情，让我感到特别开心！

# 目录

## 第 I 部分 硝烟中的XP和Scrum

第1章 简介 .....	3
免责声明 .....	4
撰写本书的原因 .....	4
Scrum到底是什么 .....	4
第2章 我们怎样编写产品backlog .....	7
额外的故事字段 .....	9
我们如何让产品backlog停留在业务层次上 .....	9
第3章 我们怎样准备sprint计划 .....	11
第4章 我们怎样制定sprint计划 .....	13
为什么产品负责人必须参加 .....	14
为什么不能在质量上让步 .....	15
无休止的sprint计划会议 .....	16
sprint计划会议日程 .....	17
产品负责人如何对sprint放哪些故事产生影响 .....	20
团队怎样决定把哪些故事放到sprint里面 .....	21
定义“完成” .....	28
使用计划扑克做时间估算 .....	29
明确故事内容 .....	30
确定每日例会的时间地点 .....	33
最后界限在哪里 .....	33