# 推荐序——David J. Anderson

“Scrum很难。”Scrum方法创始人之一Ken Schwaber这样说。Scrum是一个得到定义并且含有规范规则集的过程，这些规则集被认识到需结合在一起发挥作用。众所周知, scrum 在其设计针对的环境中起作用。对于那些采用 scrum 的人来说, 真正的挑战是改变他们的环境, 使 scrum 对他们有效。改变你的业务环境和你的客户需求并不容易——因此，“Scrum很难”。

回想起2002年，那时我第一次考虑大家现在所知道的看板方法，Scrum还没有像今天这样成为主导性的敏捷方法。敏捷那时还是个新事物，“敏捷软件开发”这个词也才被人们认识不到一年。当时最有名的敏捷方法是极限编程。许多软件开发组织仍然在使用上个世纪八九十年代制定的方法，使用诸如用例这样的技术来捕捉需求。利用甘特图来做项目计划，阶段-门限（phase-gate）管控过程强制将整个大规模项目分“阶段”进行，这些在那时都是司空见惯的。有需求收集阶段、系统分析阶段、设计阶段、等等。那时，使用小交付增量和迭代方式进行编码和测试的敏捷方法令人望而生畏，敏捷方法的普及也很缓慢。我见到最大的阻力在于采用更加宽松的需求定义方法，比如用户故事的使用；还在于去除阶段门限管控过程，这些管控过程通常是由项目集/项目组合管理办公室控制，或者是由经常向PMO汇报的审计或质量职能部门所控制。

我得出结论，如果要让全球大量的技术研发企业获得真正的业务敏捷性，我们就要使用一种新的改进方法。与其试图使用新的（却令人生畏）的敏捷方法来取代现有的方法，我们倒不如追求某种进化方式，让我们能够“从你现在正在做的开始”，并寻找其中存在的具体问题，然后针对具体问题寻求具体的解决方法。

在2004年前，我看到一种模式，IT部门和产品研发团队会过早、过多地承诺工作；制定计划往往基于理想化的假设；并且无法按承诺交付，导致客户不悦。这些失败的成因成了一种普遍的模式。所承诺过的工作会在范围和定义上变化，而新工作还会到来，进行中的工作常常由于信息收集、返工、解决生产问题等原因被迫中断。承诺的在制品被置于一旁，出于各种理由被阻塞，不足够的风险管理计划使得这些问题发生。这许多问题的解决方案是采用一套看板系统作为演进式改进过程的第一步。看板系统鼓励我们延迟承诺、限制在制品，帮助我们规避那许多常见的、导致承诺被打破、导致项目无法如期交付的问题。

如下两件事情是分开的：一是我们需要一种进化的方法来改善IT工作和产品开发中的服务交付，二是对于不切实际的过早承诺、中断问题、范围变更和增长问题以及因为各种理由造成的工作阻塞等问题来说，看板系统是一个优秀的解决方法。事实证明，需要使用看板系统的模式非常普遍，到2007年，这些概念已经有效地融入了我们现在所认为的“看板方法”的单一管理方法。

因此看板是“我们剩下人的敏捷力”。看板方法旨在为那些在21世纪不能或不愿采用这些已定义敏捷方法的人带来敏捷力。这与Scrum有什么关系？ 那么，Scrum很难。 对于渴望找到应对交付挑战的简单解决方案的热切听众来说，它可能已经超卖了。许多公司已经采用了Scrum，而没有完全满足Scrum的全部要求 - 他们并没有将其环境改变为真正适合Scrum的环境。实际上，在许多情况下，根本不可能控制你的业务环境，相反, 你的客户和你的市场为你做。他们不会为了适应你想工作的方式而改变，相反，你必须改变来适应他们期望你与他们开展业务的方式。这让世界充满了各种陷入困境的、不完整的或者黯然无效的scrum实施。

事实看起来，尽管冲刺增量更小，但是许多Scrum实施仍然遭遇到了我在十几年前的传统项目管理和软件开发生命周期方法上观察到的毛病。因此, 采用看板是一个适当的方法以帮助一个受到挑战的scrum部署实施来进化并调整适应到使用它的环境。这不会让纯粹主义者开心。这样做就不是按照已定义规则集的Scrum”，所以根据Ken Schwaber的观点，这不是Scrum。有个问题，你必须问问自己：“我们的目标是不是要完美地实施Scrum并成为全球Scrum社区的头等公民？还是说我们的目标是为了我们的顾客而改进我们的IT工作、软件开发和新产品的服务交付？” 如果你把后者看作是你的真正目标, 那么你必须接受某种程度的剪裁来开发一个对你自己的处境和 "适合于目的" 的过程。你捡起这本书并费心阅读前言的事实表明, 你已经认识到严格遵循Scrum的规则不是为你工作的, 你需要帮助来进化出更好地与你的业务环境相匹配的东西。

Ajay Reddy是一名经验丰富的Scrum实践者，你会发现与他有相同的经历。他也发现需要帮助，并且他发觉，看板方法所传递的信息与他产生了共鸣，使他能够在处理看似棘手的问题时，给出解决办法。如果Scrum没有为你工作，可能问题并不在你，也不在于你能否正确遵循那些严格规定。或许真相是你需要一种不同的过程方案，能根据你的情况量体裁衣。让看板方法帮助你达到这个目标，请允许Ajay指导你应该怎么做。

如果只有一条消息我希望你从前言中带走, 它就是在我自己的书《看板方法：科技企业渐进变革成功之道》中的话：“你有尝试看板的权利，你有权修改你的流程，你有权不一样，你的情况是独一无二的, 你应当开发一个根据你的领域、你的价值流、你管理的风险、你团队的技能以及你客户的需要而裁剪和优化得到的独一无二的流程定义”。为你迄今使用Scrum所取得的成就感到骄傲, 并拥抱这样的想法：在未来你能做得更多。让看板帮助你到达那里。

—David J. Anderson

于 华盛顿 斯奎姆

# 推荐序——Jim Benson

如果有一条精益的核心信条但却从来不曾为人们所谈及，那它肯定就是如下这条：

你必须关注你的项目。

这是多么简单。为了不断地改进，或者提供高质量的产品, 我们需要切实明白,作为一个团队, 我们正在做什么和为什么。

当我们放弃我们的项目或我们对客户的理解, 而追求规范的管理技术,那我们说我们 (我们的团队和我们的产品) 只是另一个如牵磨一样的软件工厂，那是可预知的、机械的、缺乏创造性的、愚蠢的。

但极少有项目能够声称达到了这一庸俗的境界。软件创造能够获得盈利，只因为它的差异性。差异性需要愿景。愿景需要创造力。而创造力需要实验与想象力。

我们不是机器人。软件开发不是装配流水线——远远不是。因此，软件开发人员和那些身处这一行业的人们都有一些关键的共同特征：

1. 他们是问题解决者；
2. 他们是发明家；
3. 他们是手艺人；
4. 当被禁止成为1-3时，他们真的会很生气。

在2008年，我的同事Corey Ladas写了《Scrumban: Essays on Kanban Systems for Lean Software Developement》，他通过提供两件重要的东西对日渐增多的规范的敏捷技术进行延伸：

* 用来关注的工具包
* 用来关注的权限

软件开发是一种创建其它系统的系统。它是错综复杂的。它是社会性的。我们不会因为它简单而拥有许多架架的书来讲如何开发软件。每个项目都有很多潜在的实现路径，许多潜在的生命周期，以及许多决策。

反过来，我们所创建的每个网站、APP或嵌入式程序等其自身就是一个系统。它与用户交互，它与其他软件交互，它停止工作，它需要得到维护。我们的软件产品运行在社会、金融和数码世界中，即使我们无法理解，也至少必须心存感激欣赏那些精心匠造的代码。

所以我必须诚实地对你和你当前的敏捷实践发问：

* 你是否为你的开发团队打造了一套系统，允许他们真正地看到他们工作的实时上下文？
* 如果存在问题，你的系统允许你的团队尽早地注意到吗？
* 你的系统是否允许他们不只是改变产品特性，还允许改变你的过程？
* 你的系统是否真的允许他们保持可持续的步调（开发高质量的产品），或者你是否总是尝试提升你们的速率（提升利用率）？

要做到其中任何一件事，我们都无法将开发人员从设计中分离出来。当前流行的各种版本的敏捷技术都包含了将开发人员从产品愿景中分离出来的机制。我们使用敏捷的初衷是解放程序员。然而，我们兜了一圈回到了原点。Scrum Master成了项目经理；产品负责人成了产品经理，我们在Scrum的外衣下再造了瀑布。

看板方法、个人看板方法、Scrum（类型A、B、C）、Scrumban、SAFe、RUP、XP…… 没有一个是答案。没有单独的答案。如果有，那么很明显，所有人都会那样做。单独看，这些方法与预先包装好的微波食品是一样令人担忧的健康程度。但只有将这些方法与其他技术一同烹制，你才能做出一桌真正健康的饭菜。

如果没有Kent Beck、Eliyahu Goldratt、Taiichi Ohno以及W. Edwards Deming，那么就不会有今天的看板。据David、Corey和我在开展看板中所积累经验本身是学习与关注的产物—将众多理念结合成一个。

就像一本东京驾驶员手册不会确切地告诉你如何在阿克伦开车，同样，前面提到的这些方法与理念就算是在单独地使用情况下也不会确切地告诉你如何驾驭你的项目。要想驾驭，就必须要关注。要想关注，你必须时刻看到你的工作并减少分心。

精益和看板方法可以帮助你做这些事情，但它们是令人有挫败感的。

为什么呢？

因为你和你的团队很可能处于超负荷工作状态而没有时间，或者甚至没有权限关注或者改变你的工作环境。因此，开放式的精益系统很难开始。你需要为你的团队成员、你的老板、以及其他干系人提供明确的工作方向和工作描述。谁有时间做呢？

这本书因此应运而生。

Ajay在Corey的理念上更进了一步，为那些想要成功、而不想成为罐装过程的盲目追随者的团队提供了更多潜在的实现方法。这本书将帮助你关注，来看看你正在做的工作，向别人展示你所担负的工作量，并开始对你的在制品进行控制。但不要就此止步：Ajay特意称它为进化[革命]（[R]Evolution）。进化是一个持续的过程。

这本书仿佛是隧道尽头的第一缕光亮。但请记住，只看到了光亮远远不够：你需要不断前行，不断改进你的过程，不断创造和维护尽可能最好的产品。

软件从来不眠。

——Jim Benson

新乡奖著作《Personal Kanban: Mapping Work | Navigating Life》的作者

华盛顿 海岸市

# 前言

尽管Scrumban经过多年演变已经形成为框架，但至今还没有一个明确的参考指南或者定义。事实上，正如本书前面强调的，一些“权威”资料并不同意Scrumban所实际代表的东西。Scrumban应当变得更好。

本书的目的是将一大堆混淆的矛盾的表达转变为全面、连贯、实用的资源，为希望在自己的环境中驾驭 Scrumban 力量的个人和组织带来价值。本书所讲并非某种新发展出的思想，而是推演说明一系列经证明有效的精益和敏捷原则如何能够在一个单独框架上下文中有效地得以交融和管理。同样，本书也不是一本教科书。尽管有不少建议和实践提示，但本书的主旨是指导个人和组织以有效支持其最终目的的方式来扩展他们的思路。

## 如何使用本书

阅读和掌握本书有两种不同的策略。在此之前，我们先来考虑读者的不同水平。

新入行的从业人员希望从中获得可以实施和遵从的具体步骤。

如果是处于“守”阶段的读者，Scrumban呈送了2个独特的挑战。首先，它要求从业人员要专注于系统思维，要承认一点：我们身为这些系统中的一员，如果无法首先通盘理解这些系统，就无法改进它们。其次，尽管Scrumban的一些实践和原则确实帮助定义出一些具体的步骤，但大多数不这样做。从结果看，初学者被要求拥抱的具体步骤代表了积木增量式积累到更大更深刻的理解。

现在，针对这两种策略，如何选择取决于你打算从本书中获得什么：

如果感兴趣的是对整体大图景进行理解，希望全面掌握Scrumban（这是我强烈提倡的），就请从头到尾仔细阅读全书。如果觉得自己读到了无关紧要的内容或者越读越困，就干脆略过不读。但我不建议你经常这么干，全面理解整本书对于新手而言尤为重要，这能帮你避过很多常见的陷阱。

如果对某个特定话题感兴趣，希望进行进一步了解，可以直接翻到所在章节。每一章在一个宽泛的话题上陈述了自足完整的知识，但各章是按照连贯的Scrumban之旅顺序排列的。

本书穿插了一些故事，讲述团队和组织如何使用Scrumban。在某些情况下，团队或组织会快速对Scrum的角色和实践加以掌握。在其他情况下，Scrumban作为另一种的通向敏捷的路径（这意味着选择这个路径的团队不会再实践Scrum）。两种方案最后都会得到务实而有效的成果——团队和组织对此最为看重。如果你不关心这些，那么这本书可能并不适合你。

尽管本书主要聚焦于软件开发，但Scrumban框架可以广泛应用于很多企业职能，它可以促进共同的语言和共同的理解，可以有效集成人力来齐心协力开展工作。

该框架的这一方面与整个企业中规模化精益和敏捷实践相关的挑战紧密相关, 这最终使得Scrumban 如此强大。

## 本书是如何组织的

本书分为四大部分，每部分由若干章组成。

### 第一部分 介绍

第1章 呈现：掀开Scrumban的神秘面纱

概述Scrumban的起源以及当前精益和敏捷社区对Scrumban的理解。

### 第二部分 基础——怀着终点开始

第2、3、4章旨在向读者整体概述Scrumban的根源及采用它的理由。这几章总体上呈现的材料与所有阶段的读者都相关（守、破、离三个阶段，我在第1章将对此进一步解释），尤为适合经理和高级管理者阅读。由于在每个层面打下深刻根本的理解能驱动得到更好的成效，因此我建议团队层面的个人都不要轻视这几章，特别是第2章。

第2章 秩序与混乱：从此开始

本章介绍一些关键注意事项，在探索改进时，这些事项常被忽视。我将查看经常会影响IT组织表现的历史性因素（经常影响着IT组织的表现），探究为何敏捷转型（通常指把Scrum导入组织）的努力会遭遇熄火和失败，以及如何构建Scrumban才能克服这些障碍。

第3章 任务：理清目标、价值和绩效之间的关系

当四大关键领域齐心协力，组织才能达到最成功的状态。在本章中，我着重讲述Scrumban将如何改善组织内各业务职能间的协调一致。

第4章 动机：Scrumban行之有效的秘诀

为什么用Scrumban？这一章将抽丝剥茧，进一步介绍几个核心的原则和实践，Scrumban因此而成为一种可提高员工满意度和创造高绩效的健壮框架。

### 第三部分 执行——将Scrumban付诸实践

真金不怕火炼，相对于“为什么”，第5、6、7章着重于“怎么做”。因为我强烈不建议新手在对“为什么”知之甚少的情况下，贸然开始学习“怎么做”，所以读者会发现在在实践指导性的内容中，仍然嵌入了很多解释性文字。

第5章 动员起来：推行Scrumban

详细介绍了一种在团队或组织中启动Scrumban的方式。

第6章 方法：工作机理内幕

详细解释应如何将Scrumban的特定价值观、原则和方法运用于采用的不同阶段。

第7章 度量：获得洞见与跟踪进展

本章将介绍一些指标，我们能够依赖获得新视角并使得预测更加可靠。

### 第四部分 改善——高阶话题与实践

第8、9、10章属于高阶思维——属于专业选修内容。

由于只有对Scrumban有实践经验才能理解这些内容，因此缺少经验的从业人员可能不大理解所讨论话题的重要性和影响力。不过即便如此，这些内容仍然对许多可选项和能力提供了愿景，而这些可选择性与能力会随着时间流逝被Scrumban框架暴露出来。

第8章 管理：管理是正确地做事——领导力是做正确的事

虽然Scrumban不一定非要自上而下驱动才会取得成功，但本章还是要看看为什么要持续保持长期高绩效的精髓在于优秀的领导力。

第9章 变得成熟：Scrumban如美酒，越陈越好

本章介绍帮助团队继而走向成熟的方法，以及可能会遇到的常见演变。

第10章 建模：大胆地走少有人走的路

介绍一些更高阶的技术，可以用来实现更精准的预测和适应性管理。

附录 更多：致敬勇敢的心

附录中包含了定义和附加的参考材料。

## 本书约定

我运用几个简单的约定来帮助读者顺利阅读各类信息。例如，有些话题特别适合高管和进行项目组合或项目集管理中的角色阅读，所以我就在介绍这些话题的时候会加上高管和管理人员类型的图示（这些图示是从GetScrumban借用的，这是个我参与创建的在线游戏，用做培训工具）。

类似的，Sanjay教练也将指引你留意特别重要的概念，根据在帮助很多团队在他们的环境中使用Scrumban所积累的经验，提醒你注意陷阱和误区，或给出提示。

我还力求让你知道何时需要学习更高阶话题，以便当你的理解随着时间成长的同时，能够选择回顾这些话题。

# 致谢

如果没有其他很多人的贡献和支持，本书是不可能完成的。具体说来，我要想以下人士表示感谢。

我亲爱的妻子Diana，我的四个孩子，我的母亲Susheela，他们慷慨无私的大力支持使得这本书最终得以顺利出版。

感谢CodeGenesys的好友和同事Jack Speranza，感谢他帮助我完成本书。作为CodeGenesys的首席运营官，他从教育学角度提出的观点使得本书在内容上可以满足更多读者的需求。

感谢我的好友兼搭档Marc Hughes以及我在ScrumDo的团队，感谢他们为支持本书而提供的珍贵数据和分析。

感谢CodeGenesys和ScrumDo.com的客户们，感谢他们允许我分享他们的故事。

感谢CodeGenesys的其他同僚，尤其是Haritha和Bernard，感谢他们的支持和帮助。

感谢Dimitar Bakardzhiev和Frank Vega宝贵的反馈意见。

感谢David Anderson、Jim Benson和Russell Healy的善意支持。

感谢Scrumban之父Corey Ladas，显然，没有他，就没有这本书。

还要感谢精益和敏捷社区的很多成员，他们人同意我们采撷他们的部分工作，而且他们一直都在为社区繁荣做贡献。

—Ajay Reddy

马萨诸塞州波士顿

# 关于作者

**Ajay Reddy**在过去十多年中，一直致力于帮助技术团队和组织改进工作方式。他的特点是注重实践和协作实验，核实的业务成果，提升的团队满意度。

Ajay的职业生涯之初是一名软件工程师，并因此而首次接触到诸如极限编程等敏捷方法。不久后，他便转型开始指导其他人采用敏捷思维和实践，并基于不同环境下所面临的各种挑战持续寻求过程改进之道。

2009年，Ajay创办了精益-敏捷资询公司CodeGenesys。2010年，他合作开发了ScrumDo，这是一款以推进敏捷在业内实施为明确目的的Scrum和看板方法管理工具。在过去五年时间里，他曾帮助大大小小的很多组织使用Scrumban来取得了巨大的成功。Ajay深信，Scrumban是一个特别简单但有巨大潜力的强大框架。除了写作本书，他近日来还参与创建了GetScrumban游戏，做为一种实用和有效的辅助手段，来帮助个人和组织直面Scrumban框架带的真正威力。

作为ScrumDo的首席产品策划和头牌教练，Ajay正在帮助145个国家和地区的团队和组织认识到Scrumban所能带来的红利。他在美国、印度和欧洲讲授Scrumban课程，也经常在全球各地的Scrum和看板聚会与大会上发表演讲。

2012年，Ajay悟透框架之争，认为它过于强调框架而忽略了这些框架要服务于人力系统和业务考虑的创建初衷。他相信，相对于任何框架本身而言，更为重要的是把框架当作工具为获得期望的结果提供支持。正是因为意识到这点，直接促使他花了两年时间写成了这本书。

Ajay Reddy与他的妻子Diana及四个孩子居住在马萨诸塞州。你可以在Linkedin、codegenesys.com、[www.scrumdo.com](http://www.scrumdo.com)上联系到他，或者在Twitter上@ajrdy。