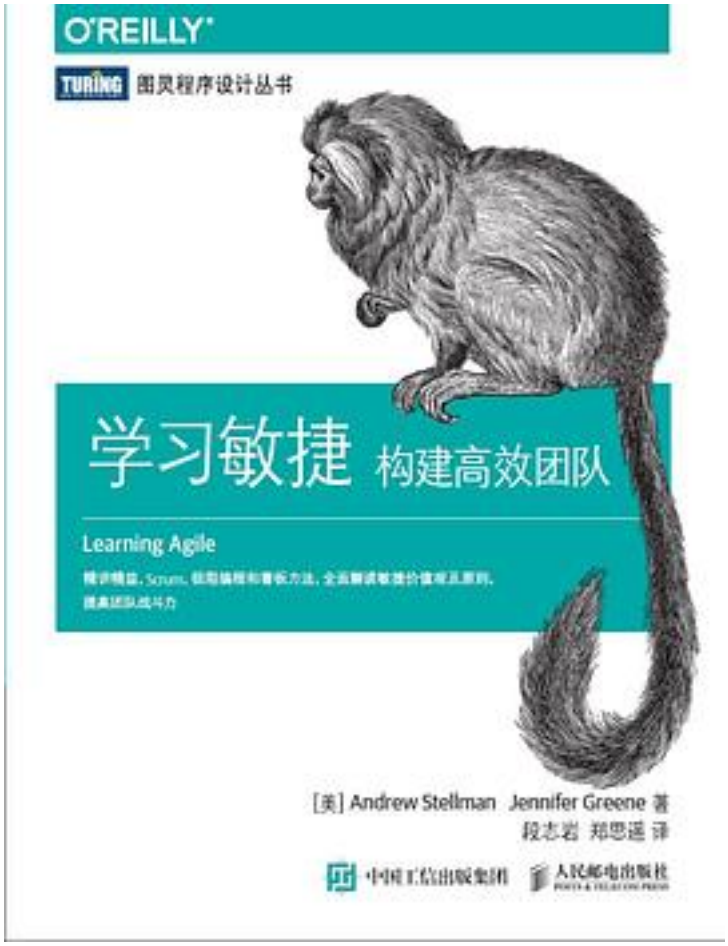
学习敏捷



学习敏捷_下载链接1_

著者:[美] Andrew Stellman

出版者:人民邮电出版社

出版时间:2017-3

装帧:平装

isbn:9787115447555

本书以敏捷软件开发为中心，系统阐述了敏捷原则和实践的先进理念和重要意义，并分别讲解了Scrum、极限编程、精益和看板四套敏捷实践的应用。作者从开发团队的日常困境入手，用讲故事的形式展开问题，由表及里，层层讲解，并在每一章最后附上参考

书，便于读者进一步查找学习。本书内容生动，语言通俗易懂，集趣味性和实用性于一体，是学习敏捷开发、提升团队效率的极佳参考书。

作者介绍:

Andrew Stellman

是O'Reilly畅销书作者、敏捷教练、项目领导人、软件工程项目经理、开发人员和系统架构师。具有20多年的软件开发项目管理经验，是公认的软件开发专家。

Jennifer Greene

是一位优秀的软件测试人员，曾与不同的软件开发团队共事，并且构建了很多相当酷的工程。她还是一位畅销书作者，曾撰写过Head First PMP、Head First C#。其中Head First C#为她与Andrew Stellman合著。

目录: 序 xv
前言 xvii
第1章 学习敏捷 1
1.1 什么是敏捷 2
1.2 本书的读者对象 5
1.3 本书的目标 6
1.4 努力建立敏捷思维 6
1.5 本书结构 9
第2章 理解敏捷价值观 11
2.1 团队主管、架构师和项目经理走进了一间酒吧…… 12
2.2 没有银弹 14
2.3 敏捷可以拯救乱局吗 16
2.3.1 引入敏捷，带来变化 17
2.3.2 “聊胜于无”的结果 18
2.4 视角割裂 19
2.4.1 视角割裂带来的问题 21
2.4.2 为什么视角割裂只能做到“聊胜于无” 22
2.5 敏捷宣言帮助团队认识实践的目的 24
2.5.1 个体和互动高于流程和工具 25
2.5.2 可工作的软件高于详尽的文档 25
2.5.3 客户协作高于合同谈判 26
2.5.4 响应变化高于遵循计划 26
2.5.5 原则高于实践 27
2.6 理解敏捷的“大象” 28
2.7 着手采用一套新方法 32
第3章 敏捷原则 37
3.1 敏捷软件开发的12 条原则 38
3.2 客户总是对的吗 38
3.3 交付项目 40
3.3.1 原则1：最优先要做的是尽早、持续地交付有价值的软件，让客户满意 40
3.3.2 原则2：欣然面对需求变化，即使是在开发后期。敏捷过程利用变化为客户维持竞争优势 41
3.3.3 原则3：频繁交付可工作的软件，从数周到数月，交付周期越短越好 42
3.3.4 改进电子书阅读器团队的项目交付计划 44
3.4 沟通和合作 46
3.4.1 原则4：在团队内外，面对面交谈是最有效、也是最高效的沟通方式 48

3.4.2 原则5：在整个项目过程中，业务人员和开发人员必须每天都在一起工作 49
3.4.3 原则6：以受激励的个体为核心构建项目，为他们提供环境和支持，
相信他们可以把工作做好 51
3.4.4 在电子书阅读器项目中采用更好的沟通方式 52
3.5 项目实施——推进项目 53
3.5.1 原则7：可工作的软件是衡量进度的首要标准 53
3.5.2 原则8：敏捷过程倡导可持续开发。赞助商、开发人员和用户要能够
共同、长期维持其步调，稳定向前 54
3.5.3 原则9：坚持不懈地追求技术卓越和设计优越，以此增强敏捷的能力 55
3.5.4 改善电子书阅读器团队的工作环境 55
3.6 项目和团队的持续改进 56
3.6.1 原则10：简单是尽最大可能减少不必要工作的艺术，是敏捷的根本 56
3.6.2 原则11：最好的架构、需求和设计来自自组织的团队 57
3.6.3 原则12：团队定期反思如何提升效率，并依此调整 57
3.7 敏捷项目：整合所有原则 58
第4章 Scrum和自组织团队 62
4.1 Scrum的规则 64
4.2 第1幕：Scrum的适用条件 65
4.3 Scrum团队中每个人都要对项目负责 67
4.3.1 Scrum主管指导团队的决策 67
4.3.2 产品所有者帮助团队了解软件的价值 68
4.3.3 每个人都对项目负责 69
4.3.4 Scrum有一组自己的价值观 75
4.4 第2幕：状态更新只是社交网络的玩法 78
4.5 整个团队参与每日Scrum例会 80
4.5.1 反馈和“可见– 检查– 调整”周期 80
4.5.2 最后责任时刻 81
4.5.3 召开有效的每日Scrum例会 83
4.6 第3幕：将冲刺计划写到墙上 86
4.7 冲刺、计划和回顾会议 87
4.7.1 迭代式与增量式 87
4.7.2 冲刺成也在于产品所有者，败也在于产品所有者 89
4.7.3 可见性和价值观 89
4.7.4 计划并执行有效的Scrum冲刺 93
4.8 第4幕：尽力之后 94
第5章 Scrum计划和集体承诺 99
5.1 第5幕：出乎意料 100
5.2 用户故事、速度和普遍接受的Scrum实践 102
5.2.1 提升软件价值 102
5.2.2 以用户故事构建用户真正会用到的功能 103
5.2.3 满意条件 105
5.2.4 故事点和速度 106
5.2.5 燃尽图 108
5.2.6 通过用户故事、故事点、任务和任务板来计划并实施冲刺 111
5.2.7 广受认可的Scrum实践 115
5.3 第6幕：第一次胜利 116
5.4 回顾Scrum价值观 116
5.4.1 具体实践没有价值观也有效果（只是别管它叫Scrum） 117
5.4.2 你的公司文化与Scrum的价值观兼容吗 119
第6章 极限编程与拥抱变化 128
6.1 第1幕：开始加班 129
6.2 极限编程的主要实践 130
6.2.1 编程实践 130
6.2.2 集成实践 131

6.2.3 计划实践 132
6.2.4 团队实践 133
6.2.5 为什么开发团队抵制变化，上述实践如何提供帮助 134
6.3 第2幕：计划有变，但我们还是看不到希望 137
6.4 极限编程的价值观帮助团队改变心态 139
6.4.1 极限编程帮助开发人员学会与用户协作 141
6.4.2 开发团队的怀疑会破坏实践的效用 142
6.5 正确的思维从极限编程的价值观开始 144
6.5.1 极限编程的价值观 144
6.5.2 以善意铺就 144
6.6 第3幕：势头的变换 147
6.7 理解极限编程价值观，拥抱变化 148
6.7.1 极限编程的指导原则 149
6.7.2 极限编程指导原则可以加深对计划的理解 151
6.7.3 极限编程指导原则与实践相互促进 152
6.7.4 反馈循环 154
第7章 极限编程、简化和增量式设计 163
7.1 第4幕：再次加班 164
7.2 代码和设计 165
7.2.1 代码异味和反模式（如何判断你是不是聪明过头了） 166
7.2.2 极限编程团队主动寻找和修复代码异味 168
7.2.3 钩子、边界情况以及功能过多的代码 170
7.2.4 代码异味会增加复杂性 175
7.3 把编码和设计决定留到最后责任时刻 175
7.3.1 决然重构，偿还技术债务 177
7.3.2 持续集成，排查设计问题 179
7.3.3 避免一体式设计 180
7.4 增量式设计与极限编程的整体实践 182
7.4.1 有时间进行思考，团队才能做好工作 184
7.4.2 团队成员彼此信任并共同作出决定 186
7.4.3 极限编程的设计、计划、团队和整体实践形成了一个带动创新的系统 186
7.4.4 增量式设计与为了复用而设计 188
7.4.5 简化单元交互，系统实现增量式成长 190
7.4.6 优秀的设计源自简单的交互 190
7.5 第5幕：最终得分 192
第8章 精益、消除浪费和着眼全局 200
8.1 精益思维 201
8.1.1 你已经理解了很多精益价值观 201
8.1.2 承诺、选择意识和集合式开发 203
8.2 第1幕：还有一件事…… 207
8.3 创造英雄与神奇思维 209
8.4 消除浪费 210
8.5 加深对产品的理解 214
8.5.1 着眼全局 216
8.5.2 找到问题的根本原因 218
8.6 尽快交付 219
8.6.1 使用面积图可视化工作进度 221
8.6.2 限制进行中的工作，控制瓶颈 225
8.6.3 拉动式系统帮助团队消除约束 226
第9章 看板方法、流程和持续改进 233
9.1 第2幕：紧赶慢赶的游戏 234
9.2 看板方法的原则 236
9.2.1 找到一个出发点并由此进行实验性的演进 236
9.2.2 用户故事进去，代码出来 238

9.3 用看板方法改进流程 240
9.3.1 将工作流程可视化 241
9.3.2 限制进行中的工作 246
9.4 测量并管理流量 251
9.4.1 用CFD 和进行中工作面积图测量并管理流量 252
9.4.2 用利特尔法则控制系统的流量 259
9.4.3 用进行中工作上限管理流量，自然地创造缓冲 263
9.4.4 让过程策略明确统一 265
9.5 看板方法下自然发生的行为 266
第10章 敏捷教练 275
10.1 第3幕：还有一件事（又来了？！）…… 276
10.2 教练要理解人们为什么不想改变 277
10.3 教练要理解人们如何学习 280
10.4 教练清楚如何让一套方法起作用 284
10.5 进行敏捷指导时的原则 285
关于作者 288
关于封面 288
・・・・・・(收起)

学习敏捷_下载链接1_

标签

敏捷

项目管理

敏捷开发

scrum

软件开发

管理

软件工程

编程

评论

Agile—增量交付，迭代改善，Scrum—自组织，Lean—消除浪费

全面了解敏捷的好书。不想学习（或读的进）的同学们，将自己归类为：实践派，而对立面是：学术派。正如三年前，我到1号楼“头脑风暴”会议室参与一个会议，当时的我对“头脑风暴”的第一反应是：这个会议室的名字真好听。对其它方面的认知是无感的。我也确认，沟通如果不注重戏方的知识结构，一味地用专业术语，也起不到沟通的目的。构建高效团队最重要的是：发现问题，分析和解决问题！重点：谦卑，知道自己不知道

先打个底，以后反复的理解-实践

对敏捷理念和方法讲的比较透的一本书，虽然书中的概念和实践都已经有了比较深的了解，类似的书籍也阅读的很多，但是这本书还是给我一些新的体悟。

敏捷的四种方法：scrum、极限编程、精益、看板
看完了前两个，有需要的话再看后两个。

思想和价值观才是灵魂

在第一次接触敏捷开发开发并初步尝试后近八年，再次回顾敏捷的一些核心思想和实践，依然很有启发。而本书也有对这些年来各项目团队不那么成功的敏捷实践案例的一些反思，这是比较可贵的。

对敏捷的理解又有了深刻的认识，但同时对于推行敏捷也更悲观了。。。根本不具备敏捷文化的基础，sign

清晰的澄清了Scrum，精益，XP和敏捷教练之间的关系。团队中的视角割裂的观点很有点儿意思。

杂烩，什么都说了，什么都没有说精通，作为入门了解还行，深入知识就没有办法跟专门讲一项的相提并论了。也许是敏捷书籍的通病，同质化太严重。在理论上，敏捷方法是不会有什么花可以玩出来的。如果成为特殊例子讲解，好像一本书永远也讲不完。所以，写敏捷理论的书绝对是吃力不讨好的事情。写敏捷时间的书，太多人又远远不到可以将实践独立抽象出来，这是现今敏捷社区无法绕过的一个坎儿。

从实践到思想，从do到think，非常科学的敏捷教程！

敏捷方法值得学习

敏捷之下的高效团队，成员组合不仅是一个共同体，还要有一致的价值观和行动力，同时具备专业性、创造力与自主性缺一不可，精英团队理念。
摒弃繁琐的流程，控制成本、优化过程、强调结果，善用工具，让问题可视化，事前分析问题进而控制流程合理性，而不是事后解决问题，浪费资源与时间。

敏捷重要框架系统讲解，赞

对自己工作流程有一定地位指导意义，推荐阅读.

对于非软件工程专业的小白来讲非常不错

本书是我学习敏捷开发的一本书。srum讲的透彻，但是其他模式讲解的模模糊糊。

学习敏捷_下载链接1_

书评

学习敏捷_下载链接1_