《敏捷技能修炼》

图书基本信息

书名：《敏捷技能修炼》

13位ISBN编号：9787111395270

10位ISBN编号：7111395271

出版时间：2012-9-5

出版社：机械工业出版社华章公司

作者：Alan Shalloway,Scott Bain,Ken Pugh,Amir Kolsky

页数：204

译者：郑立,邹骏,黄灵

《敏捷技能修炼》

前言

前言尽管本书是一本技术性的书籍，但里面涉及的很多想法都是我们在NetObjectives公司的敏捷开发培训课程中迸发出来的。过去，在给学生传授如何采用Scrum或者精益方法的时候，经常有人会问我：“我们应该怎样做，才有能力一步步地构造我们的软件呢？”答案对我而言显而易见。他们其实真正想问的问题是：“我们怎么才能用最好的方式，学习如何一步步地构造我们的软件？”关于这个问题，有下面三种方法：读书我相信任何一个读过并且读懂了《DesignPatternsExplained:ANewPerspectiveonObject- OrientedDesign》和《EmergentDesign:TheEvolutionaryNatureofProfessionalSoftwareDevelopment》的人，都知道如何一步步地写软件。参加培训这个方法要更好一点。如果能够把NetObjectives公司的课程(设计模式（DesignPatterns）和浮现式设计(EmergentDesign))结合在一起上，那就无法超越了。学习一些“小舵板”（trimtab）软件开发中的“小舵板”使得一步步地构造软件更加有效。第一种方法需要投入很多时间，第二种方法则花费很高。相比之下，第三种方法所需要的投入则小得多。遗憾的是，这些“小舵板”不是简简单单就可以讲清楚的。“什么是小舵板（trimtab)呢？”它们是飞机和船只上的零件，用以减少控制飞机副翼或者船舵所需要的能量。但是这里“小舵板”的意思则是来源于BuckyFuller曾经提到过的东西。在思考一个小小的人类个体能够做什么的时候，一些东西曾经给过我很深的冲击。想象一下“玛丽皇后”号——当整艘船远去，它的舵浮现眼前。你会发现，在舵的边缘，有一个小零件，叫做“小舵板”。它是一个微缩版的船舵。只需移动这块小小的舵板，产生一个轻微的压力，就可以拉动整个船舵转动，毫不费力。我认为小小的个体也可以成为“小舵板”。一个人在社会中的行为，就好比一艘船的舵板，细微的行动也可以驱动社会这艘大船向前行驶。所以，请叫我“小舵板”。换言之，这里要学习的“小舵板”，就是指可以让我们在较小投入下获得最深刻理解的措施和真知灼见。在设计模式课程中，我们识别出三个基本的“小舵板”。实践了这三点的学生，他们都看到了自己在设计和编程能力上的巨大提高。“这三点是什么？”当然就是本书要讲的内容：意图导向编程（ProgrammingbyIntention）。把使用从构造中分离。编码前考虑可测试性。这三点非常简单，实践它们也几乎不会增加额外的时间。实际上，它们都是与封装相关的。第一点和第三点封装的是行为的具体实现，而第二点则显然着重于封装如何构造。这个主旨非常重要，因为对实现进行封装是一种抽象过程。它是众多实现方式中的一种——在未来，可能还会有其他的方式。我认为，把新代码集成到原有系统时遇到严重问题的主要原因，就是因为忘记了这一点。要推荐的第四个“小舵板”是，遵循Shalloway原则。这一点需要多花点时间，但总是很有用处。本书就是“小舵板”的汇集。这些“小舵板”都是NetObjectives公司的讲师和教练发现的敏捷软件开发人员可以遵从的基本要素，告诉我们如何以有效的形式来编写高质量的代码。你可以在闲暇的时间里，按任何的章节顺序来阅读本书。也就是说，这些章节顺序如此编排仅仅是为了协助我们理顺思绪而已。

《敏捷技能修炼》

内容概要

《敏捷技能修炼：敏捷软件开发与设计的最佳实践》的4位作者都是世界顶级的软件开发专家和敏捷导师，都有数十年的软件行业从业经验，其中3位曾荣获Jolt大奖。本书是敏捷软件开发领域公认的经典著作，权威性毋庸置疑。
书中内容围绕“敏捷式编程”这一主题展开，对每一位敏捷软件开发人员都应该掌握的核心技能和技术进行了深入阐述，总结出了大量最佳实践，提供了一整套最精炼的技术集合，可以帮助他们在开发中变得游刃有余，极大地提高开发效率和软件质量。
《敏捷技能修炼：敏捷软件开发与设计的最佳实践》共分四个部分：第一部分（1~7章），阐述了在软件开发过程中能起到“四两拨千斤”作用的几种思想方法（“小舵板”），如意图导向编程、分离构造和使用、测试先行和Shalloway原则等，并总结了业界常用的几种实践，包括如何封装、面向接口的设计和验收测试驱动等；第二部分（8~9章），对过度设计和持续集成这两个问题进行了深入的探讨，并给出了最佳实践；第三部分（10~13章），作者分享了很多只有在他们的教学现场才能获得的经验，这些经验是优秀架构师应该具备的，具体包括共性和可变性分析、以开放关闭原则为目标的重构、需求与功能接口、何时以及如何使用继承等重要内容；第四部分是附录，介绍了统一建模语言、提高代码质量的原则，以及如何封装原始数据类型等。

《敏捷技能修炼》

作者简介

Alan Shalloway，Net Objectives公司创始人及CEO。Alan有40多年工作经验，他是计算机软件行业，特别是精益、看板、产品系列管理、scrum和敏捷设计方面的思想领导者。他帮助公司在企业级层面向精益和敏捷方式转型，同时教授员工相关课程。Alan开发了关于精益－敏捷的培训辅导方法，这帮助Net Objectives公司的客户取得了长期的可持续的生产力。他经常活跃于全球范围的高端峰会并发表精彩演讲。他还是《Design Patterns Explained： A New Perspective on Object-Oriented Design》（Jolt获奖作品）和《Lean-Agile Pocket Guide for Scrum Teams》的主要作者。在他的职业生涯中，Alan从事过多个行业。他是精益软件和系统协会（the Lean Software and Systems Consortium）的共同创始人及董事会成员。他拥有麻省理工学院计算机科学系的硕士学位和艾莫利大学（Emory University）数学系的硕士学位。更多详细信息请查看Twitter上的@alshalloway。

Scott Bain，在计算机技术方面有将近40年经验，从事过软件开发、软件工程、框架设计等方面的工作。Scott也曾经从事课堂和远程教学等教育活动，包括课程的设计、实施培训和相关管理，给用户提供认证培训和终端用户培训。目前Scott在敏捷分析和设计模式、高级软件设计和可持续的测试驱动开发等方面进行授课和提供咨询。同时，Scott还经常在JavaOne和SDWest这样的开发者研讨会上进行精彩的演讲。他是《Emergent Design： The Evolutionary Nature of Professional Software Development》的作者，此书荣获了Jolt生产力奖。

Ken Pugh， Net Objectives公司高级咨询师。凭借多年的丰富经验，他提供培训和辅导，帮助公司向精益－敏捷转型。他热衷于研究沟通（特别是有效地传递需求）、业务价值交付，以及用精益原则来快速进行高质量的交付。同时，在技术课题方面，他提供从面向对象设计到Linux/Unix等多方面的培训和指导。此外，他还写过好几本编程书籍，包括获得2006年Jolt大奖的《Prefactoring： Extreme Abstraction， Extreme Separation， Extreme Readability》。 最近的一本书是《Lean-Agile Acceptance Test Driven Development： Better Software Through Collaboration》。他的客户遍及伦敦、波士顿、悉尼、北京和海得拉巴。工作之余，他喜欢滑雪、帆船、自行车和阿巴拉契亚徒步登山活动。

Amir Kolsky，Net Objectives的资深咨询师、教练和培训师。Amir从事计算机科学领域已经超过25年了。他在IBM研究院工作过10年，此外有9年时间在各种大小类型的公司担任过首席架构师和首席技术官等职位。他从2000年开始接触敏捷开发。他先后创建了MobileSpear 以及XPand软件公司，专门在以色列和欧洲提供敏捷辅导、软件教育和敏捷项目实施。目前Amir把他的专业经验带到了Net Objectivs，作为敏捷教练和讲师，提供关于精益和敏捷软件流程、工具和实践、Scrum、极限编程、设计模式以及测试驱动开发方面的培训。

《敏捷技能修炼》

书籍目录

《敏捷技能修炼》

章节摘录

《敏捷技能修炼》

编辑推荐

《敏捷技能修炼:敏捷软件开发与设计的最佳实践》编辑推荐：4位世界顶级软件开发专家、敏捷导师兼Jolt大奖获得者数十年工作经验结晶，敏捷软件开发领域公认的经典著作。围绕意图导向编程、分离构造和使用、测试先行、Shalloway原则、面向接口设计、测试驱动开发、避免过度设计、持续集成、共性和可变性分析、重构等核心技术主题给出了大量最佳实践，字字珠玑。

《敏捷技能修炼》

精彩短评

1、里面的思想很好！
2、十条，木有最佳实践
3、不是能全部理解，是自己知识面不够宽广，项目经验不够丰富，工作中遇到问题会再回头研读体会。
4、挺好 的 理论知识挺不错
5、还行，该公司技穷于此矣
6、简单翻了一下，还不错，就是感觉书薄了一点
7、有敏捷实践的基础，看起来会容易很多。但是讲的很浅显很泛。
8、已经开始按照书中介绍的一些原则和方法在自己的代码中一步步进行实践
9、很多实用性强的原则和方法，非常值得学习
10、书的内容单薄，不值得这个定价。
11、书内容不错，就是太薄了
12、这是一个非常好的书,总结出了很多最佳实践
13、:
TP311.521/3132
14、不错，值得一看，但有点难懂。
15、看了一下这本书，发现很多技巧都是通用的.在其它的书中也提到过这些。
16、比较关键的工程实践都覆盖到了，写的也是深入浅出，翻译也可以。
17、打好自己的基础,看看前辈的指导.
18、大师写的值得一看
19、书中分析了多种开发者必备的软件开发技能，就像一本《九阳真经》，至于能练到哪一重，就靠自己勤学苦练了:D
20、挺好，给老公带的，应该实用
21、很多人，或许强调敏捷的一个结果，快速原型。但是本质上的敏捷反而从来不去实践，为实现持续集成需要的架构，自动化工具和实践，测试驱动开发，意图导向的编码，还有敏捷的原则。本书没有过多探讨团队，但是其实强大的团队和流程才是敏捷的核心
22、还可以，感觉对日常工作帮助不是太大
23、对于OO初学者还是非常有用的。我通过本书也了解了更多OO的知识。遗憾的是，我比较期待的意图导向编程与共性可变性分析讲得太简单了。
24、设计和代码层面的敏捷方法
25、都是实践性很强的内容，对于认真思考过软件设计的码农来说浅显易懂， 书不厚
26、好是好，难题在工作中怎么用
27、敏捷开发领域的一本好书
28、开完第一章，感觉我还没到看这书层次，还是看看模板去
29、在各种敏捷技能已经成为common sense的今天, 这本书对其中重要的部分总结的很好.
30、中等的推荐
31、书不厚， 提出的理念也不算多，不过都是值得好好领会、实践和掌握的。 有些内容略显单薄了些， 对不住4位顶级软件开发专家的名头， 总体还是可以的，

《敏捷技能修炼》

精彩书评

1、抛开人云亦云的敏捷不谈，这本书通过引入一些例子和概念，来引导大家去思考如何进行高质量的软件设计。读这本书，会使我一直沉浸在印证自己的编程经验的状态中，时而，恍然大悟，时而，追悔莫及。读这本书，你会像我一样有所收获，不过更重要的是坚持不懈的实践，实践出真知。
2、《敏捷技能修炼：敏捷软件开发与设计的最佳实践》的4位作者都是世界顶级的软件开发专家和敏捷导师，都有数十年的软件行业从业经验，其中3位曾荣获Jolt大奖。本书是敏捷软件开发领域公认的经典著作，权威性毋庸置疑。书中内容围绕“敏捷式编程”这一主题展开，对每一位敏捷软件开发人员都应该掌握的核心技能和技术进行了深入阐述，总结出了大量最佳实践，提供了一整套最精炼的技术集合，可以帮助他们在开发中变得游刃有余，极大地提高开发效率和软件质量。《敏捷技能修炼：敏捷软件开发与设计的最佳实践》共分四个部分：第一部分（1~7章），阐述了在软件开发过程中能起到“四两拨千斤”作用的几种思想方法（“小舵板”），如意图导向编程、分离构造和使用、测试先行和Shalloway原则等，并总结了业界常用的几种实践，包括如何封装、面向接口的设计和验收测试驱动等；第二部分（8~9章），对过度设计和持续集成这两个问题进行了深入的探讨，并给出了最佳实践；第三部分（10~13章），作者分享了很多只有在他们的教学现场才能获得的经验，这些经验是优秀架构师应该具备的，具体包括共性和可变性分析、以开放关闭原则为目标的重构、需求与功能接口、何时以及如何使用继承等重要内容；第四部分是附录，介绍了统一建模语言、提高代码质量的原则，以及如何封装原始数据类型等。

《敏捷技能修炼》

版权说明