第一图书网, tushu007.com

<<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

出版时间：机械工业出版社华章公司

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

页数：185

译者：郑立,邹骏,黄灵

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

第一图书网, tushu007.com

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。

第一图书网, tushu007.com

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。

第一图书网, tushu007.com

<<敏捷技能修炼>>

内容概要

本书的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。

全书共分四个部分：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。

第一图书网, tushu007.com

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。

第一图书网, tushu007.com

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。

第一图书网, tushu007.com

<<敏捷技能修炼>>

书籍目录

推荐序
译者序
丛书前言
前言
致谢
第一部分最关键的小舵板
第1章意图导向编程
1.1意图导向编程：一个实例
1.2优点
1.2.1方法的内聚性
1.2.2可读性和表达性
1.2.3调试
1.2.4重构和增强
1.2.5单元测试
1.2.6更易修改和扩展
1.2.7在代码中发现模式
1.2.8可迁移的方法
1.3小结
第2章分离构造和使用
2.1一个重要的问题
2.2两种视图
2.2.1创建视图
2.2.2使用视图
2.2.3隐藏的部分更容易改动
2.2.4现实的做法
2.2.5一些实际的考量因素
2.3给你的决策计时
2.4重载和C++
2.5自我查验
2.6小结
第3章代码未动，测试先行
3.1一个小舵板：测试与可测试性
3.2什么是测试
3.3可测试性和代码质量
3.4案例学习：可测试性
3.4.1随时应对变化
3.4.2青蛙一样的程序员
3.5一个关于测试先行的思考
3.5.1更好的设计
3.5.2更清晰的范围和避免不必要的工作
3.5.3降低复杂性
3.5.4其他优势
3.5.5没有例外
3.6小结
第4章Shalloway法则和Shalloway原则

第一图书网, tushu007.com

4.1冗余的种类
4.1.1复制和粘贴
4.1.2“ 魔法 ”数字
4.1.3其他类型
4.2重新定义冗余
4.3其他形式的冗余
4.4设计模式在减少冗余时扮演的角色
4.5很少有开发人员花费大量的时间去“ 修改 ”代码错误
4.6冗余对代码质量其他方面的影响
4.7小结
第5章封装
5.1未封装的代码：对全局变量的破坏
5.2成员标志的封装
5.3自封装成员
5.4预防代码更改
5.5封装引用对象的难点
5.6用get()来打破封装
5.7对象类型的封装
5.8设计的封装
5.9各个层次的封装
5.10实用性建议：把困难封装起来
5.11小结
第6章面向接口的设计
6.1针对接口的设计
6.2接口的定义
6.3接口约定
6.4分离不同的视图
6.5接口的模拟实现
6.6让接口保持简单
6.7避免过早采用继承体系
6.8接口和抽象类
6.9依赖反转原则
6.10多态性概述
6.11不是每个类都需要接口
6.12小结
第7章验收测试驱动开发
7.1两种开发流程
7.2验收测试
7.3一个关于验收测试的实例
7.4实现验收测试
7.4.1针对用户界面的测试脚本
7.4.2测试用户界面
7.4.3XUnit测试
7.4.4验收测试框架
7.4.5四种方法间的联系
7.5一个练习
7.6如果客户不告诉你怎么做的时候，你应该怎么办

第一图书网, tushu007.com

7.7小结
第二部分基本态度
第8章避免过度设计或设计不足
8.1给开发人员的箴言
8.2代码质量病理学
8.3避免过度设计或设计不足
8.4把复杂度和返工最小化
8.5永不把代码变得更糟/仅在有目的的情况下降低代码质量
8.6使代码容易修改，足够强大健壮，适应变化并安全可靠
8.7在非面向对象的代码或遗留系统里编写易于修改代码的策略
8.8小结
第9章持续集成
9.1建立源代码分支
9.1.1多版本：特殊分支
9.1.2孤立地工作：开发分支
9.1.3问题、解决方案、新的问题
9.2将主干内容合并回分支
9.3测试驱动开发与合并成本
9.4持续集成
9.5持续集成服务器
9.6小结
第三部分设计问题
第10章共性和可变性分析
10.1用动词和名词来做指南：警告，前面有危险
10.2真正的问题是什么
10.3我们所需要知道的
10.4共性和可变性分析
10.4.1共性分析
10.4.2可变性分析
10.4.3面向对象设计“一箭三雕”
10.5发掘对象的新范式
10.6分析矩阵：一个用例学习
10.7小结
第11章以开放关闭原则为目标的重构
11.1开放关闭原则
11.1.1从开放关闭原则引申到其他
11.1.2开放关闭原则是一个“原则”
11.2重构
11.2.1为何重构
11.2.2负债还是投资
11.2.3重构和遗留系统
11.2.4以开放关闭原则为目标的重构
11.2.5“及时”设计
11.3小结
第12章需求与功能接口
12.1迪米特法则
12.2耦合，可恶的耦合，还有依赖

第一图书网, tushu007.com

12.2.1耦合和可测试性
12.2.2需求与功能
12.3理想的分离方案：需求接口和
功能接口
12.4回到迪米特法则
12.5小结
第13章何时以及如何使用继承
13.1“四人组”
13.2初始向量，最终结果
13.3优先委托
13.4使用继承与使用委托
13.5继承的使用
13.6可扩展性
13.7在敏捷开发里应用四人组的训诫
13.8测试问题
13.9更多
第四部分附录
附录A统一建模语言概览
附录B代码质量
附录C封装原始数据类型

第一图书网, tushu007.com

章节摘录

第一图书网, tushu007.com

<<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。

第一图书网, tushu007.com

<<敏捷技能修炼>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com