

<<敏捷软件开发>>

图书基本信息

书名：<<敏捷软件开发>>

13位ISBN编号：9787115165077

10位ISBN编号：7115165076

出版时间：2008-1

出版时间：人民邮电

作者：马丁

页数：732

字数：922000

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>

## <<敏捷软件开发>>

### 内容概要

本书旨在指导.NET程序员学习构建软件的最佳实践，从而创建更好的设计并提升.NET应用的质量。书中使用真实案例讲解如何用极限编程来设计、测试、重构和结对编程，包含了极具价值的可重用的C#源代码，还重点讲述了如何使用UML和设计模式解决面向客户系统的问题。

本书适于软件开发和管理人员提高自身水平学习之用，也适于用作高校计算机专业本科生、研究生以及软件学院的软件工程和软件开发相关课程的教材或参考书。

## 作者简介

Robert C.Martin世界级的软件开发大师，著名软件咨询公司Object Mentor公司的创始人和总裁。曾经担任C++ Report杂志主编多年，也是设计模式和敏捷开发运动的主要倡导者之一。

## &lt;&lt;敏捷软件开发&gt;&gt;

## 书籍目录

Section I : Agile Development 敏捷开发 Chapter 1 : Agile Practices 敏捷实践 The Agile Alliance 敏捷联盟  
 Individuals and Interactions over Processes and Tools人和交互重于过程和工具 Working Software over Comprehensive Documentation 可以工作的软件重于面面俱到的文档 Customer Collaboration over Contract Negotiation 客户合作重于合同谈判 Responding to Change over Following a Plan 随时应对变化重于遵循计划 Principles 原则 Conclusion 结论 Bibliography 参考文献  
 Chapter 2 : Overview of Extreme Programming 极限编程概述 The Practices of Extreme Programming 极限编程实践 Whole Team 完整团队 User Stories 用户故事 Short Cycles 短交付周期  
 Acceptance Tests 验收测试 Pair Programming 结对编程 Test . Driven Development(TDD) 测试驱动开发 Collective Ownership 集体所有权 Continuous Integration 持续集成  
 Sustainable Pace 可持续的开发速度 Open Workspace 开放的工作空间 The Planning Game 计划游戏 Simple Design 简单设计 Refactoring 重构 Metaphor 隐喻 Conclusion 结论  
 Bibliography 参考文献 Chapter 3 : Planning 计划 Initial Exploration 初始探索 Spiking , Splitting , and Velocity 探究、分解和速度 Release Planning 发布计划 Iteration Planning 迭代计划  
 Defining “ Done ” 定义“完成” Task Planning 任务计划 Iterating 迭代 Tracking 跟踪  
 Conclusion 结论 Bibliography 参考文献 Chapter 4 : Testing 测试 Ttest-Driven Development 测试驱动开发  
 Example of Test-First Design 测试优先设计的例子 Test Isolation 测试促使模块之间隔离 Serendipitous Decoupling 意外获得的解耦合 Acceptance Tests 验收测试  
 Serendipitous Architecture 意外获得的构架 Conclusion 结论 Bibliography 参考文献 Chapter 5 : Refactoring 重构  
 A Simple Example of Refactoring : Generating Primes 素数产生程序：一个简单的重构示例 Unit Testing 单元测试 Refactoring 重构 The Final Reread 最后审视  
 Conclusion 结论 Bibliography 参考文献 Chapter 6 : A Programming Episode 一次编程实践 The Bowling Game 保龄球比赛  
 Conclusion 结论 Overview of the Rules of Bowling 保龄球规则概述 Section II Agile Design 敏捷设计  
 Chapter 7 : What Is Agile Design 什么是敏捷设计 Design Smells 设计臭味 Design Smells—The Odors of Rotting Software 设计坏味——腐化软件的气味  
 Rigidity 僵化性 Fragility 脆弱性 Immobility 顽固性 Viscosity 粘滞性 Needless Complexity 不必要的复杂性  
 NeedLless Repetition 不必要的重复 Opacity 晦涩性 Why Software Rots 软件为何会腐化 The Copy Program Copy 程序 A Familiar Scenario 熟悉的场景  
 Agile Design of the Copy Program copy 程序的敏捷设计 Conclusion 结论 Bibliography 参考文献 Chapter 8 : The Single-Responsibility Principle(SRP) SRP : 单一职责原则  
 Defining a Responsibility 定义职责 Separating Coupled Responsibilities 分离耦合的职责 Persistence 持久化 Conclusion 结论  
 Bibliography 参考文献 Chapter 9 : The Open/Closed Principle(OCP) OCP : 开放-封闭原则 Description Of OCP OCP概述  
 The Shape Application Shape 应用程序 Violating OCP 违反OCP Conforming to OCP 遵循OCP  
 Anticipation and “ Natural ” Structure 预测变化和“贴切的”结构 Putting the “ Hooks ” In 放置吊钩  
 Using Abstraction to Gain Explicit Closure 使用抽象获得显式封闭 Using a Data-Driven Approach to Achieve Closure 使用“数据驱动”的方法获取封闭性  
 Conclusion 结论 Bibliography 参考文献 Chapter 10 : The Liskov Substitution Principle(LSP) LSP : Liskov替换原则  
 Violations Of LSP 违反LSP的情形 A Simple Example 简单例子 A More Subtle Violation 更微妙的违反情形  
 A Real-World Example 实际的例子 Factoring Instead of Deriving 用提取公共部分的方法代替继承  
 Heuristics and Conventions 启发式规则和习惯用法 Conclusion 结论 Bibliography 参考文献 Chapter 11 : The Dependency-Inversion Principle(DIP) DIP : 依赖倒置原则  
 Layering 层次化 Ownership Inversion 倒置的接口所有权 Dependence on Abstractions 依赖于抽象  
 A Simple DIP Example 简单的DIP示例 Finding the Underlying Abstraction 找出潜在的抽象  
 The Furnace Example 熔炉示例 Conclusion 结论 Bibliography 参考文献 Chapter 12 : The Interface Segregation Principle(ISP) ISP : 接口隔离原则  
 Interface Pollution 接口污染 Separate Clients Mean Separate Interfaces 分离客户就是分离接口 Class Interfaces versus

## &lt;&lt;敏捷软件开发&gt;&gt;

Object Interfaces 类接口与对象接口      Separation Through Delegation 使用委托分离接口  
 Separation Through Multiple Inheritance 使用多重继承分离接口      The ATM User Interface Example  
 ATM用户界面的例子      Conclusion 结论      Bibliography 参考文献      Chapter 13 : Overview of UML  
 for C# Programmers C#程序员UML概观      Class Diagrams 类图      Object Diagrams 对象图  
 Sequence Diagrams 顺序图      Collaboration Diagrams 协作图      State Diagrams 状态图      Conclusion  
 结论      Bibliography 参考文献      Chapter 14 : Working with Diagrams 使用UML      Why Model? 为什么  
 建模      Why Build Models of Software? 为什么构建软件模型      Should We Build Comprehensive  
 Designs Before Coding? 编码前应该构建面面俱到的设计吗      Making Effective Use of UML 有效使  
 用UML      Communicating with Others 与他人交流      Road Maps 脉络图      Back-End  
 Documentation 项目结束文档      What to Keep and What to Throw Away 要保留的和要丢弃的  
 Iterative Refinement 迭代式改进      Behavior First 行为优先      Check the Structure 检查结构  
 Envisioning the Code 想象代码      Evolution of Diagrams 图的演化      When and How to Draw  
 Diagrams 何时以及如何绘制图示      When to Draw Diagrams and When to Stop 何时要画图, 何时不要  
 画图      CASE Tools CASE工具      But What About Documentation? 那么, 文档呢      Conclusion  
 结论      Chapter 15 State Diagrams 状态图      The Basics 基础知识      Special Events 特定事件  
 Superstates 超状态      Initial and Final Pseudostates 初始伪状态和结束伪状态      Using FSM Diagrams  
 使用FSM图示      Conclusion 结论      Chapter 16 : Object Diagrams 对象图      A Snapshot in Time 即时快  
 照      Active Objects 主动对象      Conclusion 结论      Chapter 17 : Use Cases 用例      Writing Use Cases  
 编写用例      Alternate Courses 备选流程      What Else? 其他东西呢      Diagramming Use Cases 用例  
 图      Conclusion 结论      Bibliography 参考文献      Chapter 18 : Sequence Diagrams 顺序图      The Basics  
 基础知识      Objects, Lifelines, Messages, and Other Odds and Ends 对象、生命线、消息及其他  
 Creation and Destruction 创建和析构      Simple Loops 简单循环      Cases and Scenarios 时机和场  
 合      Advanced Concepts 高级概念      Loops and Conditions 循环和条件      Messages That Take  
 Time 耗费时间的消息      Asynchronous Messages 异步消息      Multiple Threads 多线程      Active  
 Objects 主动对象      Sending Messages to Interfaces 向接口发送消息      Conclusion 结论      Chapter 19  
 : Class Diagrams 类图      The Basics 基础知识      Classes 类      Association 关联      Inheritance  
 继承      An Example Class Diagram 类图示例      The Details 细节      Class Stereotypes 类衍型  
 Abstract Classes 抽象类      Properties 属性      Aggregation 聚集      Composition 组合  
 Multiplicity 多重性      Association Stereotypes 关联衍型      Nested Classes 内嵌类  
 Association Classes 关联类      Association Qualifiers 关联修饰符      Conclusion 结论      Bibliography  
 参考文献      Chapter 20 Heuristics and Coffee 咖啡的启示      The Mark IV Special Coffee Maker Mark IV  
 型专用咖啡机      Specification 规格说明书      260      A Common but Hideous Solution 常见的丑陋方案  
 Imaginary Abstraction 虚构的抽象      An Improved Solution 改进方案      Implementing the  
 Abstract Model 实现抽象模型      The Benefits of This Design 这个设计的好处      Ooverkill 面向对象过  
 度设计      Bibliography 参考文献      Section      The Payroll Case Study 薪水支付案例研究  
 Rudimentary Specification of the Payroll System 薪水支付系统的初步规格说明      Exercise 练习  
 Use Case 1 : Add New Employee 用例1 : 增加新雇员      Use Case 2 : Deleting an Employee 用例2 : 删  
 除雇员      Use Case 3 : Post a Time Card 用例3 : 登记考勤卡      Use Case 4 : Posting a Sales Receipt  
 用例4 : 登记销售凭条      Use Case 5 : Posting a Union Service Charge 用例5 : 登记工会服务费  
 Use Case 6 : Changing Employee Details 用例6 : 更改雇员明细      Use Case 7 : Run the Payroll for  
 Today 用例7 : 现在运行薪水支付系统      Chapter 21 : Command and Active Object : Versatility and  
 Multitasking      COMMAND模式和ACTIVE OBJECT模式 : 多功能与多任务      Simple Commands 简单  
 的Command      Transactions 事务      Physical and Temporal Decoupling 实体上解耦和时间上解耦  
 Temporal Decoupling 时间上解耦      Undo Method Undo()方法      Active Object ACTIVE  
 OBJECT模式      Conclusion 结论      Bibliography 参考文献      Chapter 22 : Template Method and Strategy  
 : Inheritance versus Delegation      TEMPLATE METHOD模式和STRATEGY模式 : 继承和委托  
 Template Method      TEMPLATE METHOD模式      Pattern Abuse 滥用模式      Bubble Sort 冒泡排

<<敏捷软件开发>>

序 Strategy STRATEGY模式 Conclusion 结论 Bibliography 参考文献 Chapter 23 : Facade and Mediator FACADE模式和MEDIATOR模式 Facade FACADE模式 Mediator MEDIATOR模式 Conclusion 结论 Bibliography 参考文献 Chapter 24 : Singleton and Monostate SINGLETON模式和MONOSTATE模式 Singleton SINGLETON模式 Benefits SINGLETON模式的好处 Costs SINGLETON模式的代价 Singleton in Action 运用SINGLETON模式 Monostate MONOSTATE模式 Benefits MONOSTATE模式的好处 Costs MONOSTATE模式的代价 Monostate in Action 运用MONOSTATE模式 Conclusion 结论 Bibliography 参考文献 Chapter 25 : Null Object NULL OBJECT Description 描述 Conclusion 结论 Bibliography 参考文献 Chapter 26 : The Payroll Case Study : Iteration 1 薪水支付案例研究 : 第一次迭代开始 Rudimentary Specification 初步的规格说明 Analysis by Use Cases 基于用例分析 Adding Employees 增加新雇员 Deleting Employees 删除雇员 Posting Time Cards 登记考勤卡 Posting Sales Receipts 登记销售凭条 Posting a Union Service Charge 登记工会服务费 Changing Employee Details 更改雇员明细 Payday 发薪日 Reflection : Finding the Underlying Abstractions 反思 : 找出底层的抽象 Employee Payment 雇员支付类别抽象 Payment Schedule 支付时间表抽象 Payment Methods 支付方式 Affiliations 从属关系 Conclusion 结论 Bibliography 参考文献 Chapter 27 : The Payroll Case Study : Implementation 薪水支付案例研究 : 实现 Transactions 事务 Adding Employees 增加雇员 Deleting Employees 删除雇员 Time Cards , Sales Receipts , and Service Charges 考勤卡、销售凭条以及服务费用 Changing Employees 更改雇员属性 What Was I Smoking? 犯了什么晕 Paying Employees 支付雇员薪水 Paying Salaried Employees 支付领月薪的雇员薪水 Paying Hourly Employees 支付钟点工薪水 Main Program 主程序 The Database 数据库 Conclusion 结论 About This Chapter 关于本章 Bibliography 参考文献 Section IV : Packaging the Payroll System 打包薪水支付系统 Chapter 28 : Principles of Package and Component Design 包和组件的设计原则 Packages and Components 包和组件 Principles of Component Cohesion : Granularity 组件的内聚性原则 : 粒度 The Reuse/Release Equivalence Principle(REP) 重用-发布等价原则 The Common Reuse Principle(CRP) 共同重用原则 The Common Closure Principle(CCP) 共同封闭原则 Summary of Component Cohesion 组件内聚性总结 Principles of Component Coupling : Stability 组件的耦合性原则 : 稳定性 The Acyclic Dependencies Principle(ADP) 无环依赖原则 The Stable-Dependencies Principle(SDP) 稳定依赖原则 The Stable-Abstractions Principle(SAP) 稳定抽象原则 Conclusion 结论 Chapter 29 : Factory FACTORY模式 A Dependency Problem 依赖问题 Static versus Dynamic Typing 静态类型与动态类型 Substitutable Factories 可替换的工厂 Using Factories for Test Fixtures 对测试支架使用对象工厂 Importance of Factories 工厂的重要性 Conclusion 结论 Bibliography 参考文献 Chapter 30 : The Payroll Case Study : Package Analysis 薪水支付案例研究 : 包分析 Component Structure and Notation 组件结构和符号 Applying the Common Closure Principle(CCP) 应用CCP Applying the Reuse/Release Equivalence Principle(REP) 应用REP Coupling and Encapsulation 耦合和封装 Metrics 度量 Applying the Metrics to the Payroll Application 度量薪水支付应用程序 Object Factories 对象工厂 Rethinking the Cohesion Boundaries 重新思考内聚的边界 The Final Packaging Structure 最终的包结构 Conclusion 结论 Bibliography 参考文献 Chapter 31 : Composite COMPOSITE模式 Composite Commands 组合命令 Multiplicity or No Multiplicity 多重性还是非多重性 Conclusion 结论 Chapter 32 : Observer : Evolving into a Pattern OBSERVER——演化至模式 The Digital Clock 数字时钟 The Observer Pattern OBSERVER模式 Models 模型 Management of OOD Principles 面向对象设计原则的运用 Conclusion 结论 Bibliography 参考文献 Chapter 33 : Abstract Server, Adapter, and Bridge ABSTRACTSERVER模式、ADAPTER模式和BRIDGE模式 Abstract Server ABSTRACT SERVER模式 Adapter ADAPTER模式 The Class Form of Adapter 类形式ADAPTER模式 The Mode,m Problem , Adapters , and LSP 调制解调器问题、适配器以及LSP Bridge BRIDGE模式 Conclusion 结论 Bibliography 参考文献 Chapter 34 : Proxy and Gateway : Managing Third-Party APIs PROXY模式和GATEWAY模式 : 管理第三方API Proxy PROXY 模式

Implementing Proxy 实现PROXY模式      Summary 小结      Databases , Middleware , and Other  
 Third . Party Interfaces 数据库、中间件以及其他第三方接口      Table Data Gateway TABLE DATA  
 GATEWAY      Testing and In . Memory TDGs 测试和内存TDG      Testing the DB Gateways 测  
 试DbGateWay      Using Other Patterns with Databases 可以用于数据库的其他模式      Conclusion 结论  
 Bibliography 参考文献Chapter 35 : Visitor VISITOR模式      VISITOR VISITOR模式      Acyclic Visitor  
 ACYCLIC VISITOR模式      Uses of VISITOR 使用VISITOR模式      Decorator DECORATOR模式  
 Extension Object EXTENSION OBJECT模式      Conclusion 结论      Bibliography 参考文献Chapter 36  
 : State STATE模式      Nested Switch/Case Statements 嵌套switch/case语句      The Internal Scope State  
 Variable 内部作用域的状态变量      Testing the Actions 测试动作      Costs and Benefits 代价和收益  
 Transition Tables 迁移表      Using Table Interpretation 使用表解释      Costs and Benefits 代价和收益  
 The State Pattern STATE模式      State versus Strategy STATE模式和STRATEGY模式      Costs and  
 Benefits 代价和收益      The State Machine Compiler(SMC) 状态机编译器      Turnstile.CS Generated by  
 SMC , and Other Support Files SMC生成的Turnstile.CS以及其他支持文件      593      Costs and Benefits 代  
 价和收益      Classes of State Machine Application 状态机应用的场合      High-Level Application Policies for  
 GUIs 作为GUI中的高层应用策略      GUI Interaction Controllers GUI交互控制器      Distributed  
 Processing 分布式处理      Conclusion 结论      Bibliography 参考文献Chapter 37 : The Payroll Case Study  
 : The Database 薪水支付案例研究 : 数据库      Building the Database 构建数据库      A Flaw in the Code  
 Design 一个代码设计缺陷      Adding an Employee 增加雇员      Transactions 事务      Loading an Employee  
 加载Employee对象      What Remains? 还有什么工作      Chapter 38 : The Payroll User Interface : Model  
 View Presenter 薪水支付系统用户界面 : Model-View-Presenter      The Interface 界面      Implementation  
 实现      Building a Window 构建窗口      The Payroll Window Payroll 窗口      The Unveiling 真面目  
 Conclusion 结论      Bibliography 参考文献Appendix A : A Satire of Two Companies 双公司记      Rufus  
 Inc . : Project Kickoff      Rufus公司 : “日落”项目      671      Rupert Industries : Project Alpha      Rupert工业  
 公司 : “朝晖”项目      Appendix B : What Is Software? 什么是软件Index索引

## &lt;&lt;敏捷软件开发&gt;&gt;

## 编辑推荐

《敏捷软件开发：原则、模式与实践(C#版)(英文版)》不仅是一部深入浅出、生动易懂的面向对象原则与模式著作，而且还是一部通俗的敏捷方法导引书和快速实用UML教程。通过《敏捷软件开发：原则、模式与实践(C#版)(英文版)》你会发现，许多以前看起来非常枯燥费解的概念，忽然间都豁然开朗，变得鲜活生动起来。

C#版与此前的Java版相比，主要的更新包括加强了对UML的介绍章节，使其更加贴近实战，增加了对MVP模式的介绍等。

注释版在原著基础上增加了丰富的词语注释和背景提示，使读者能够轻松地领略原汁原味的名著的风采。

软件开发的不朽经典，生动阐述面向对象原则、敏捷实践、UML和模式，大量C#实战示例，让你亲历现场，丰富的词汇和背景注释，助你轻松读经典。

“我最喜爱的技术作家Robert Martin善于通过实战展示技术，让读者能够以自己喜欢的方式逐步理解……请把Bob大叔当作你在敏捷世界里的导师。

”——Chris Sells，NET资深技术专家，微软“软件传奇人物” “《敏捷软件开发：原则、模式与实践(C#版)(英文版)》是对敏捷编程和敏捷原则最全面和最有价值的介绍……绝对是所有NET程序员必读之作。

”——Jesse Liberty，微软资深技术专家，ProgrammingC#作者 要想成为一名优秀的软件开发员，需要熟练应用编程语言和开发工具，更重要的是能够领悟代美代码背后的原则和前人总结的经验——这正是《敏捷软件开发：原则、模式与实践(C#版)(英文版)》的主题。

《敏捷软件开发：原则、模式与实践(C#版)(英文版)》凝聚了世界级软件开发大师Robert C Martin数十年软件开发和培训经验，Java版曾荣获计算机图书最高荣誉——Jolt大奖，是广受推的经典著作，自出版以来一直畅销不衰。



#### 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>