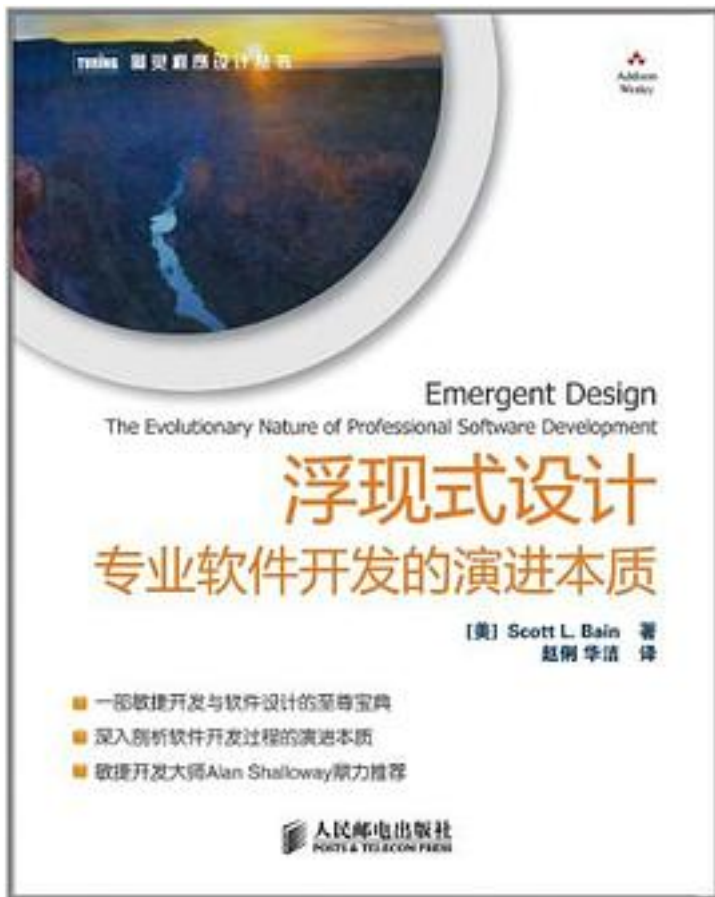


浮现式设计



[浮现式设计_下载链接1](#)

著者:Scott L.Bain

出版者:人民邮电出版社

出版时间:2011-8

装帧:平装

isbn:9787115259783

《浮现式设计：专业软件开发的演进本质》主要面向软件开发者群体，尤其是对敏捷开发感兴趣的程序设计人员。浮现式设计是一种敏捷技术，强调在开发过程中不断演进。由ScottL.Bain编著的《浮现式设计：专业软件开发的演进本质》的讨论围绕着专业软件开发方法的演进主题展开，强调了让软件成为一个真正专业的重要性，以及以演进方式

开发软件的重大意义。书中谈到了如何在演进过程中综合运用设计模式、重构、单元测试和测试驱动开发等实践，以及何时制定耦合、内聚和封装等关键决策，而且通过准确生动的示例说明了如何开发出真正有用的软件。

作者介绍:

拥有30年从业经验的资深计算机技术专家，主要从事开发、工程和设计。他还曾负责设计、提供和管理认证培训课程，从事最终用户的技能培训，既有课堂授课，也有远程教育。过去8年来，Scott一直在华盛顿州普捷湾畔的Net Objectives公司工作，负责培训课程以及有关设计模式、重构、单元测试、测试驱动开发的咨询工作。Scott与Net Objectives的CEO Alan Shalloway一起为敏捷环境中的设计模式集成做出了卓越的贡献。他还经常在各种开发者会议（例如JavaOne和SDWest）上发表演讲。

目录: 第1章 软件开发这个职业	1
1.1 人类制作软件已经有多久的历史了	1
1.2 软件开发是一种什么样的活动	2
1.3 软件开发缺少了什么	4
1.4 谁说了算	6
1.5 独特性	6
第2章 从衣橱到探月	8
2.1 软件开发中的模式和专业化	8
2.2 Andrea 的衣橱	9
2.3 探月	13
2.3.1 因素的连锁变化	16
2.3.2 不同的因素导致不同的设计	16
2.3.3 还有更多环境因素	17
2.3.4 成本和获益	18
2.3.5 火星探险	18
2.4 模式的价值	19
2.5 小结	20
第3章 软件开发的本质	21
3.1 失败率过高	21
3.2 成功的定义	22
3.3 Standish Group	23
3.4 做了错误的事情	24
3.5 做事的方式错了	25
3.6 随着时间的推移，软件开发也有所改善	27
3.7 一个原因：土木工程的类比	27
3.8 放弃希望	29
3.9 有时等待和拖延也是必要的	30
3.10 桥是硬的，软件是软的	30
3.11 我们在变化的海洋中游泳	31
3.12 接受变化	31
3.13 拥抱变化	32
3.14 利用变化	32
3.15 更好的类比：不断演进的系统	34
3.16 小结	37
第4章 代码的演进：初级阶段	38
4.1 用对象结构来代替过程逻辑	38
4.2 面向对象和模式的起源	39

- 4.3 一个示例：简单条件和Proxy 模式 40
- 4.4 下一步：多路径条件选择 43
- 4.5 为什么要采用对象结构 45
- 4.6 从多个条件中选择一个 46
- 4.7 小结 46
- 第5章 使用和发现模式 48
 - 5.1 根据上下文进行设计：我做的另一个木匠活 48
 - 5.2 模式引出了另一个看问题的角度 55
 - 5.3 模式提供了一种讨论设计的语言 55
 - 5.4 本书中的模式 56
 - 5.5 小结 56
- 第6章 软件开发金字塔 58
 - 6.1 构成专业的元素 58
 - 6.2 一种形象的表示 60
 - 6.3 小结 60
- 第7章 注重软件质量 61
 - 7.1 封装 62
 - 7.2 内聚 62
 - 7.2.1 方法内聚 63
 - 7.2.2 视角层的内聚 65
 - 7.2.3 类内聚 66
 - 7.2.4 内聚到何种程度才足够 67
 - 7.3 耦合 67
 - 7.3.1 有意耦合与意外耦合 68
 - 7.3.2 耦合类型 69
 - 7.4 冗余 73
 - 7.5 可测试性 77
 - 7.6 可读性 78
 - 7.7 软件的病症 79
 - 7.7.1 内聚性较差的信号 79
 - 7.7.2 意外耦合或不合逻辑耦合的信号 80
 - 7.7.3 冗余的信号 81
 - 7.8 小结 81
- 第8章 注重原则和智慧结晶 83
 - 8.1 使用与创建分离 83
 - 8.1.1 Fowler 的三层视角 83
 - 8.1.2 另一种视角 84
 - 8.1.3 使用的视角 85
 - 8.1.4 一个单独的视角：创建 86
 - 8.1.5 最后考虑构造细节 87
 - 8.1.6 回到现实 88
 - 8.2 开闭原则 89
 - 8.2.1 类级的开闭原则 90
 - 8.2.2 方法级的开闭原则 91
 - 8.3 依赖倒置原则 92
 - 8.4 GoF 的建议 93
 - 8.4.1 设计方法的接口 93
 - 8.4.2 设计类的接口 94
 - 8.4.3 GoF：优先使用对象聚合而非类继承 95
 - 8.5 GoF：在设计中思考什么应该变化并封装会发生变化的概念 98
 - 8.6 小结 100
- 第9章 注重实践 101
 - 9.1 统一编码风格 101
 - 9.1.1 注释 102

- 9.1.2 命名类、方法和变量 103
- 9.1.3 编码标准的好处 104
- 9.2 意图导向编程 105
- 9.3 封装构造函数 107
 - 9.3.1 原则与实践 110
 - 9.3.2 做出决定 110
- 9.4 公共性—可变性分析 111
- 9.5 实践与自由 114
- 9.6 小结 115
- 第10章 注重纪律：单元测试 116
 - 10.1 测试的经济学 116
 - 10.1.1 单元测试 117
 - 10.1.2 先写测试 119
 - 10.2 JUnit 框架 120
 - 10.2.1 JUnit 基础知识 121
 - 10.2.2 JUnit 示例 122
 - 10.2.3 Rule.java：先编码，再测试 122
 - 10.2.4 RuleContainer.java：先测试，再编码 128
 - 10.2.5 消除冗余：@Before和@After 135
 - 10.2.6 自动化批量测试 137
 - 10.2.7 异常和单元测试 139
 - 10.3 模拟对象 141
 - 10.3.1 MockObject 框架 142
 - 10.3.2 伪对象 145
 - 10.3.3 依赖注入和Endo-Testing技巧 146
 - 10.3.4 Endo-Testing 147
 - 10.4 小结 148
- 第11章 注重纪律：重构 149
 - 11.1 重构质量糟糕的代码 150
 - 11.2 重构质量优秀的代码 151
 - 11.3 结构变化与功能变化 152
 - 11.4 重构可帮助你做出选择 153
 - 11.5 模式可以成为重构的目标 154
 - 11.6 避免重构：预构 154
 - 11.7 重构技巧 155
 - 11.8 重构遗留代码 162
 - 11.9 小结 164
- 第12章 测试驱动开发 165
 - 12.1 何谓测试驱动开发 165
 - 12.1.1 测试驱动与先写测试 165
 - 12.1.2 从单元测试的角度来设计 166
 - 12.2 测试与质量 167
 - 12.2.1 测试与内聚 167
 - 12.2.2 测试与耦合 168
 - 12.2.3 测试与冗余 169
 - 12.3 测试驱动开发与模式 169
 - 12.3.1 Strategy 模式 169
 - 12.3.2 乌龟站在乌龟上，一直向下 170
 - 12.3.3 模拟对象/模拟乌龟 171
 - 12.4 模拟对象 172
 - 12.5 模拟乌龟 174
 - 12.6 测试Decorator 模式 174
 - 12.7 小结 178
- 第13章 模式与因素 179

13.1 在演进的设计中做决策	179
13.2 Christopher Apexander 与他所提出的“因素”	180
13.2.1 信号处理器示例	180
13.2.2 PKZip 示例	184
13.2.3 测试与因素	186
13.3 更多选择，更多因素	187
13.4 小结	190
第14章 浮现式设计：案例分析	191
14.1 问题领域：MWave 公司	191
14.2 团队	192
14.3 最简单的能够正常运作的设计	194
14.4 新需求：更复杂的机器	196
14.5 顺便介绍一下	198
14.6 更多好消息	199
14.7 小结：设计是一次漫长而奇特的旅行	200
第15章 结束语：展望2020年	202
附录A 演进路径	204
附录B 示例中用到的模式简介	213
附录C 有用幻觉之原理	274
参考书目	279
• • • • •	(收起)

[浮现式设计 下载链接1](#)

标签

软件开发

软件设计

软件工程

敏捷开发

计算机

编程

软件架构

评论

可能是现在能力不够……总之看得一头雾水，老外作者虽然用自己在家里当木工的事例来类比软件开发，我还是很想咨询下出版社进这本书干啥……

艾玛这本书变成小组阅读材料了。靠，明天就要讨论了尼玛我才发现！幸好之前刚好读完了前三章orz直接把豆瓣笔记发出去了hiahiahia

没接触过极限编程的人估计还是觉得很新鲜的，很喜欢开篇作家具的例子和对于计算科学与医学的比较，非常幽默

万金油著作

还行吧，没什么突出之处

没有太多新东西，收获不大。测试，重构，模式，OO杂烩

重新认识模式与上下文，给了我需要的整体框架

[浮现式设计_下载链接1](#)

书评

原文首发于图灵社区。 <http://www.ituring.com.cn/article/196>

很多软件开发人员很困惑，自身的职业发展是怎样的？靠谱的公司怎么那么少？很多软件公司很困惑，软件开发人员怎么招聘，怎么培养，怎么用？为什么靠谱的软件开发人员那么少？如果你有上述疑惑，在思考职业的软件开...

书的标题很唬人,不过内容却是相对精简.

此书是作者根据自己的开发经验以及授课经验写就的.里面其实没有很高深的东西,也没有涉及到非常详细的东西.但是很多东西都是作者的经验之谈,特别是对如何运用设计模式(当然也没有一一细谈)很有自己的见解,也很实用.其实很多从业人员对...

作者凭着30多年的编程、教学经验讲解一些软件设计开发的最佳实践——设计模式、重构、TDD，再结合自己的一些个人经历说明设计模式的作用。书皮上宣传得过于夸张，起码我没有读到任何新东西。

从《设计模式》《重构》《领域设计》后，最近几年确实没有软件设计方面的大作，我们...

[浮现式设计_下载链接1](#)