

重视大脑的学习指南

Head First 软件开发

(中文版)



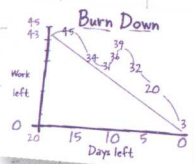
学习Mary如何满足她的客户的真实用户故事



通过开发速度弄清你的团队的生产率并以此进行评估



通过测试驱动开发去避免不可见的软件灾难



通过跟踪burn-down率保持你的项目进度



掌握经验丰富的软件开发人员的技术和工具



O'REILLY® 中国电力出版社

Dan Pilone & Russ Miles 著
陈燕国 陈荧 林乃强 译

Head First软件开发

(中文版)

我做梦都想，要是有一本软件开发的书能使我成为一个较优秀的软件开发人员，而不是让人感觉枯燥无味，该有多好！这可能只是一个幻想……



ian Pilone, Russ Miles 著

陈燕国 陈荧 林乃强 译

O'REILLY®

Beijing • Cambridge • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

Head First软件开发 (中文版) / (美) 皮隆尼 (Pitone, D.), (美) 迈尔斯 (Miles, R.) 著; 陈燕国, 陈荧, 林乃强译, —北京: 中国电力出版社, 2009. 9

书名原文: Head First Software Development

ISBN 978-7-5083-9007-9

I. H... II. ①皮... ②迈... ③陈... ④陈... ⑤林... III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第102211号

北京版权局著作权合同登记

图字: 01-2009-3669号

©2007 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2009. Authorized translation of the English edition, 2007 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2007。

简体中文版由中国电力出版社出版, 2009。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名/	Head First软件开发 (中文版)
书 号/	ISBN 978-7-5083-9007-9
责任编辑/	孙芳
封面设计/	Louise Barr, Steve Fehler, 张健
出版发行/	中国电力出版社
地 址/	北京三里河路6号 (邮政编码100044)
印 刷/	航远印刷有限公司
开 本/	880毫米×1230毫米 20开本 24.5印张 655千字
版 次/	2010年6月第1版 2010年6月第1次印刷
印 数/	0001—3000册
定 价/	68.00元 (册)

对《Head First Software Development》的高度赞扬

“《Head First Software Development》看是一本幽默滑稽的书，但实为一本精心铺设的丛书，全书充满实用的信息、有趣的图解和精辟的说明，意图是把宝贵的知识正确且清晰地植入你的大脑。它完全是让你耳目一新的书。”

— Scott Hanselman

软件开发人员，发言人，作家

Scott Hanselman's Computer Zen

“这是一本那些经验丰富的开发人员希望在他们项目开始前就读过的书。我知道我是其中之一。”

— Burk Hufnagel, 高级软件架构师

“如果在我上一个项目之前就阅读过这本书，我就能避免让全世界都能感觉到痛苦！”

— 这个开发人员要求匿名，所以她的上一个项目的项目经理就不再心烦了！

“《Head First Software Development》一书讲授了很多有价值的课程，这些课程能帮助任何一个开发人员按时间和在预算内交付高质量的软件。依据该书中讲授的核心原则将有助于你的项目自始至终在正确的轨道上。不管你已经从事软件开发的时间有多久，《Head First Software Development》将始终为你开发成功的软件提供基本的工具。”

— Adam Z. Szymanski, 软件项目经理，Naval研究实验室

“该书中的思想可被新的和有经验的项目经理使用，用以提高软件开发的全过程。”

— Dan Francis, 软件工程经理，财富50强公司

“软件开发过程的崭新视野，从需求到交付全过程管理开发团队的好介绍。”

— McClellan Francis, 软件工程师

对《Head First Software Development》一书的赞誉

《Head First Software Development》一书给人一种耳目一新的感觉，这本书突出的特点是该书把重点放在学习上。市面上有很多关于这一主题的书籍，它们都花很多时间告诉你“为什么”，但都无法让一个从业人员真正地把OOA&D落在项目之中。尽管这些书也都非常令人感到有趣，但并不实用。我深信软件开发实践的未来必然落在从业人员的身上，本书的作者们让OOA&D对从业人员不再遥不可及，并且能在从业人员的实践中发挥作用。

— Ivar Jacobson, Ivar Jacobson顾问公司

“我刚刚看完这本书，我喜爱它！该书设法通过UML和使用案例，清楚地说明面向对象的分析与设计的基本思想，甚至对良好的设计也有精辟的说明，全都是紧凑的步调和易于理解的方式进行。我最喜欢本书的一件事是该书把重点放在我们为什么要实践OOA&D——编写伟大的软件！通过定义何为伟大的软件，以及详细说明OOA&D过程中的每一步，引领读者达到那个目标。本书甚至让最疲惫不堪的Java编程人员明白为何OOA&D确实很重要。对Java新手，甚至对已经在业界工作过一段时间但饱受一些OOA&D“巨著”之苦的Java程序人员来说，这是最好的“首选之书”。

— Kyle Brown, IBM杰出工程师

“终于，一本OOA&D的好书面世了，认清UML只是辅助工具，开发软件的首要任务在于花时间让各个议题仔细想清楚”。

— Peter McBreen, 《Software Craftmanship》的作者

“本书采用了‘Head First’系列的编写风格，对于充满趣味性和可视导向的效果处理得非常好。然而，隐含在诙谐图片和趣味文字背后的是对OOA&D这一主题的严肃认真和充满智慧的艺术化的诠释。本书对如何设计程序和做有效的沟通提出了强有力的观点。我喜爱该书采用变化的例程引导读者理解设计过程的不同阶段。阅读本书时，感觉就像站在设计专家的肩膀上听他解释每一步中哪些是重要的，并能知其所以然。”

— Edward Sciore, 波士顿学院计算机科学系副教授

“这是一部精心编排的书籍，它实现了对读者的承诺：如何进行分析、设计以及编写真正的面向对象的软件。本书流畅地采用使用案例来捕捉需求以用于分析、设计、实施、测试和循环。在面向对象开发的每一步都被呈现在健全的软件工程原则之下。这些案例不仅清晰，而且具有说明性。这是一本关于面向对象软件开发，内容充实且令人耳目一新的好书。”

— Dung Zung Nguyen, Rice大学讲师

对《Head First Software Development》一书的赞誉

“我昨天刚收到这本书，而且立刻在回家的路上开始阅读，简直是爱不释手。于是，我把该书带到健身房，一边运动，一边阅读，期望人们不断看到我脸上的笑容。这样太酷了！这本书不但有趣，涵盖很多基础知识，而且观点正确，真地让我印象深刻。”

— Erich Gamma, IBM杰出工程师, 《Design Pattern》的合著者

“本书试图融生动趣味、洞察力、技术深度以及实用于一体，成为一本寓教于乐的书籍。不管你是初次学习设计模式，或者已具有多年使用设计模式的经验，你一定能从参观对象村（Objectville）的过程中学习到很多东西。”

— Richard Helm, 《Design Pattern》的合著者

“我感觉好像刚刚把千斤万担的书举过头顶。”

— Ward Cunningham, Wiki的发明者, Hillside集团的创始人

“这本书近乎完美，因为它在提供专业知识的同时，也保持着相当高的可读性。论述具有权威性，同时能轻松阅读。它是我所读过的软件书籍中，极少数让我觉得不可或缺的一本书。”

— David Gelernter, 耶鲁大学计算机科学系教授, 《Mirror World》和《Machine World》的作者

“在设计模式的王国中，复杂会变简单，简单也会变复杂，而这本书就是最好的指南。”

— Miko Matsumura, 产业分析师, SUN公司Chief Java Evangelist

“我笑，我哭，这本书深深地感动了我。”

— Daniel Steinberg, Java.net 主编

“我的第一反应是：笑得在地板上打滚。笑完之后，我意识到这本书不仅仅是技术上精湛，更是我曾经读过的有关设计模式的书籍中，最简单易懂的入门书。”

— Timothy A. Budd 博士, Oregon州立大学计算机科学系副教授, 十多本书的作者（包括《Java 程序员的C++》）

“在NFL里，Jerry Rice（译注）的花样无人能及，但本书的作者在模式设计方面更胜一筹。说真的，这本书是我曾经读过的最有趣、最聪明的书。”

— Aaron LaBerge, ESPN.com副总裁

译注：NEL：美国美式足球联盟。

Jerry Rice：前NFL明星球员。

O'Reilly其他相关的书籍

Making Things Happen

Applied Software Project Management

Beautiful Code

Prefactoring

The Art of Agile Development

UML 2.0 In a Nutshell

Learning UML 2.0

O'Reilly其他Head First系列的书籍

Head First Java

Head First Object-Oriented Analysis and Design (OOA&D)

Head Rush Ajax

Head First HTML with CSS and XHTML

Head First Design Patterns

Head First Servlets and JSP

Head First EJB

Head First PMP

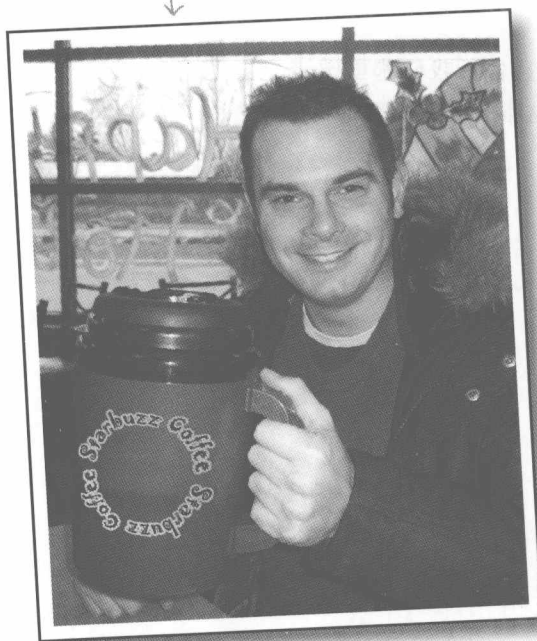
Head First SQL

Head First JavaScript

本书献给所有与我们一起做过项目，并且告诉我们哪里做错、哪里做对、应该阅读什么书的人……，这本书是我们对他们的回馈。

本书的作者

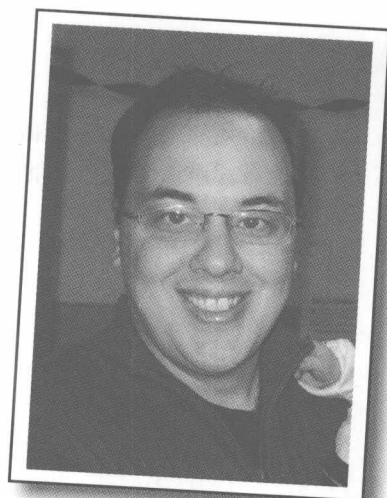
Russ Miles



Russ由衷感谢他的未婚妻Corinne，因为她的爱和支持。哦，他还不敢相信她同意下一年与他结婚，但我想有些人是幸运的！

Russ一直在著书立说，揭开了大量的技术、工具与技巧的神秘的面纱。在多年担任不同层次的软件开发人员后，Russ现正忙于带领一组开发人员为音乐界开发一些超级神秘的服务。另外，他刚刚完成了牛津大学五年的研究生课程的学习并获得了硕士学位。他一直期望有一点休息时间……，但不能如他所愿。

Russ是一个狂热的吉他演奏者，期望在业余时间能演奏吉他。他错失的唯一的一件事情就是编写《Head First Guitar》……，嘿，Brett（本书编辑），你也想要一本吧！



Dan Pilone

Dan永远感激他的妻子，Tracey，是他的妻子支持他完成了本书的编著工作。Dan是Vangent公司的软件架构师并且领导了Naval研究实验室和NASA的一个小组，Dan还构建了企业软件，他还在华盛顿特区的Catholic大学教授本科生和研究生的软件工程课程，他讲授的有些课程非常有趣。

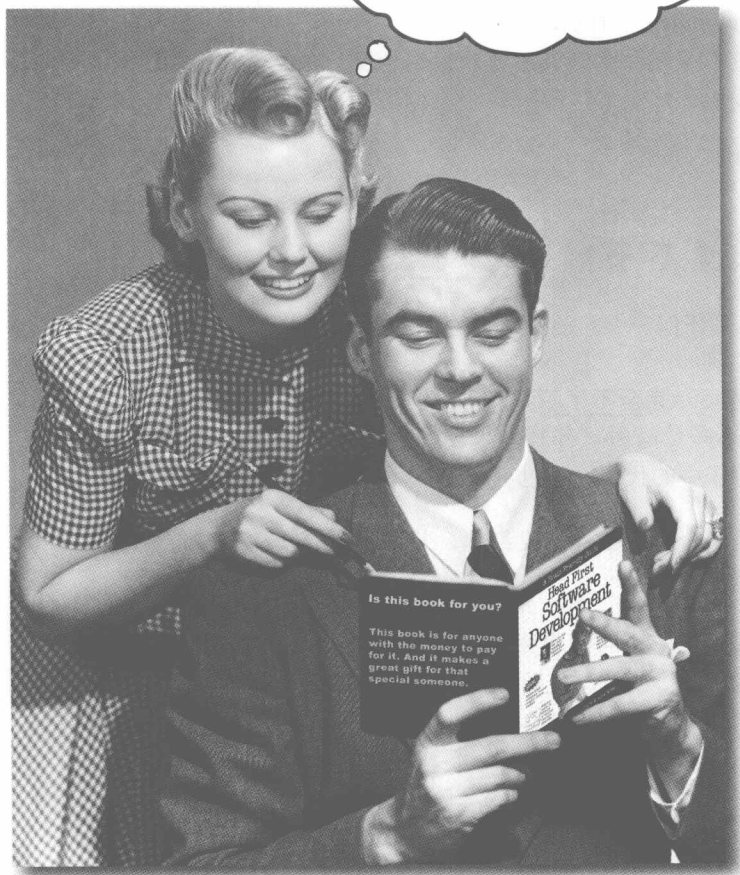
五年多前Dan就向O'Reilly提出了撰写这本书的建议。三本UML方面的书籍，在Boulder Colorado与O'Reilly的Head First团队的合作，后来是合作编著，终于他把这本书合编在一起。

虽然在领导一个软件开发团队（一项有挑战性的工作）的经验丰富，但对于如何为人父母确无能为力，Dan耐心地期盼有人能写一本名叫《Head First为人父母之道》的书，去帮助解决一些棘手的、复杂的管理问题。

如何使用本书

引言

我真不敢相信，他们把这些内容都写进了一本软件开发的书里！



在这一部分，我们回答一个热点问题：“他们为什么把这些都写进一本软件开发的书里？”

本书为谁写?

如果以下问题的全部答案都是“肯定”的:

- ① 你具有一定的计算机和一些编写程序的背景知识吗?
- ② 你想要学会构建和交付伟大软件的技术吗? 你想理解在开发循环 (Iteration) 和测试驱动开发 (Test-driven development) 背后的原则吗?
- ③ 你喜欢采用午餐会而不喜欢采用枯燥乏味的学术性的研讨吗?

本书就是为你写的。

在本书中, 我们使用Java, 但你可以假设它是C#, 然而你不会把它看作为Perl。

谁可能会放弃阅读本书?

如果你对以下任何一个问题的回答是“肯定”的:

- ① 你对Java完全不了解吗?
(你不必是高手, 并且如果你懂得C++或C#你将能理解例子中的代码, 这也可以了)
- ② 你是一个超级软件开发经理, 并且正在寻找一本参考书吗?
- ③ 你害怕尝试不同的事物吗? 你宁可接受牙医治疗, 也不愿意混搭条纹与花格子布的衣服吗? 你认为将开发循环的观念拟人化的技术性书籍是不够认真和严肃的吗?

那这本书就不是为你写的。



[来自营销部门的提示: 本书是适合任何使用信用卡的人写的]

我们知道你在想什么

“这本书怎可能是一本专业性强的软件开发书？”

“这些图是干什么的？”

“我真地能用这种方式学到点什么吗？”

我们知道你的脑子在想什么

你的脑子渴求新奇的事物，它总是在搜寻和等待不平常的事情的出现。大脑生来如此，正是这样的特点，它能有助于你保持活力。

那么，当你的大脑总是面对一成不变的流程、平淡和普通的事物时，它会怎样反映呢？大脑能做的事情是阻止这些日常的事情干扰大脑真正的工作——并记录最要紧的事情。大脑不会费心地记忆那些无聊的事情；它们也从不让“这明显的不重要”的事情通过过滤器。

你的大脑如何知道什么是重要的事情呢？假设你出去徒步旅行时，一只老虎突然跳到你的面前，在你的大脑和身体中会有怎样的反应？

中枢神经被触发了，情绪激动了，体内的化学物质激增。这就是你的大脑是如何知道的……

这一定是很重要！千万不要忘记它！

但是，你想象一下，你现在在家里或是在图书馆。这是一个安全的、温馨的，没有老虎出没的地方。你正在复习，准备考试，或者你正在研究某个你的老板认为需要一周或顶多十天内就能够弄清楚的技术难题。

然而，有一个问题。你的大脑试图去帮你的忙，并试图确保这明显的不重要的内容不要弄乱你有限的记忆。要把记忆真正地花在了要记住的大事情上，如老虎、火灾的危险，或像那个在MySpace上玩“BigDaddy”的家伙不是我要在下午六点钟要见的人。

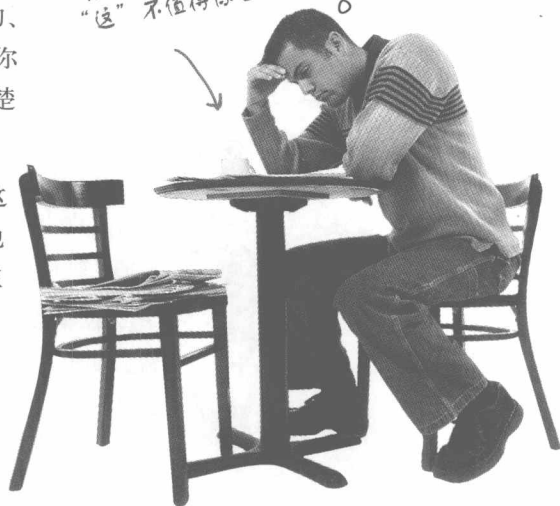
而且，也没有一个简单的方法告诉你的大脑，“嗨，大脑，拜托了，但不管这本书是多么枯燥，也不管我能刚刚注册到令人激动的Richter scale，还是请你把这些内容全部记下来。”

你的大脑认为“这”是重要的。



太好了。只有450多页沉闷的、乏味的、令人生厌的书要看。

你的大脑认为“这”不值得保留。



我们把“Head First”的读者作为学习者

那么，要怎么学会的一点东西呢？首先，你必须理解它，然后确认你不会忘记它。这不是要把一些知识塞进你的大脑。基于认知科学、神经生物学和教育心理学的最新研究，学习过程所需要的绝对不只是书本的内容。我们知道如何启发你的大脑。

Head First的学习原则：

可视化。图形比单纯的文字更容易记忆，并且使学习更为有效（在知识的记忆与转换上有接近89%的提升）。图形也能使阅读的内容更容易理解。将文字**放进或靠近他们相关联**



图形，而不是把文字放在页脚或下一页，让学习者具有接近2倍的效率来解决相关的问题。

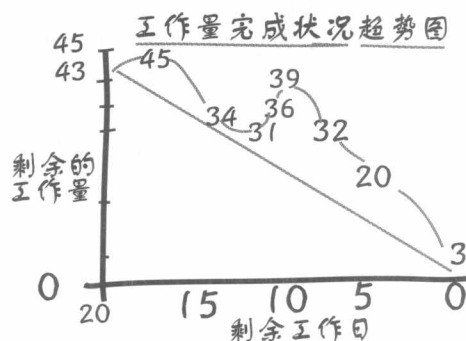
采用对话式和拟人化的风格。根据最新的研究，如果直接与读者进行对话，并且采用第一人称和会话式的风格而不是采用正儿八经的语气，学生课后测验的成绩可提升40%。用讲故事代替论述，以轻松的语言取代正式的演说。不要太严肃，想想看，是晚宴上伴侣间的窃窃私语还是课堂上死板的演说，更能吸引你的注意力？

让学习者更深入思考。换句话说，除非你主动刺激你的神经元，不然，在你的大脑中就什么也没有发生。读者必须被刺激、必须参与、受好奇心的驱使去解决问题，得出结论，并且形成新的知识。为此，你需要挑战、练习和令人深思的问题，以及包括左脑、右脑和多种感官的各种活动。

引起——并保持读者的注意力。我们都有这样的经历，就是“我真地想去学习这些，但我还没有

翻过一页，就已经昏昏欲睡了”。你的大脑只注意到了异常、有趣、奇特、吸引眼球和意想不到的事情。学会一个新的、艰深的技术问题未必是令人枯燥乏味的，如果你不觉得无聊，大脑的学习效率就会提升很多。

触动读者的情感。现在，我们知道你记忆事情的能力很大程度上取决于具有情感的内容。你会记住你在乎的事情，当你能感同身受时，你就记住了。不！我们不是在讲述一个小男孩和他的小狗之间令人心碎的故事，而是在说，当你完成了猜字游戏，学会了别人认为很难的东西时，或发现自己比别人懂得更多技术时，所产生的惊喜、好奇、有趣，“哇靠……？”和“我好棒”，诸如此类的情绪和感觉。



元认知：关于思考的思考

如果你真地想要学习，并且想要学得更快和更深入，那么，请注意你是如何“集中注意力”的。“想想”你是如何思考的。“学学”你是如何学习的。

在我们的成长过程中，大多数人都没有学习过元认知，或者有关学习理论的课程。师长期待我们去学习，但极少告诉我们如何去学习。

但是，假如你手里正拿着我们这本书，你真地想去学习如何开发伟大的软件系统，而且，可能你不想去花太多的时间。如果你想去运用在本书中读到的东西，就必须记住你学习到的内容。为此，你要理解它的内容。为了从本书中得到最多收益，或从任何书或学习经验中获得更多知识，你的大脑就要承担这方面的责任，让它好好注意这一方面的内容。

学习的技巧在于：让你的大脑去认为你正在学习的新知识确实很重要，事关你的福祉，就像吃人的老虎一样。否则，你就要不停地苦战，想要记住，确总是记不住。

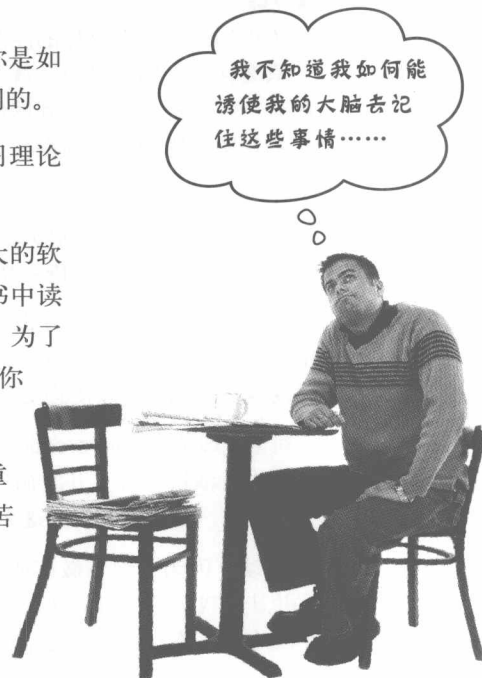
那么，要如何让你的大脑视软件开发为一只饥饿的老虎一样？

有慢而乏味的方式，也有快而有效的方式。慢的方式就是完全的反复，你肯定知道，勤能补拙，只要你不不停地向你的大脑灌输同样的学习内容，再乏味的知识，也能够学会并记住。通过足够多的反复，你的大脑会说，“这对他而言，不觉得重要，但他一遍又一遍地苦读同样的内容，所以我认为这内容应该是很重要的”。

较快的方式是去做任何提高你的大脑主动性的事情，特别是不同类型的大脑活动。上一页中的内容是解决方法的一大部分，并且已被证明能有助于大脑的有效工作。例如，研究表明把文字放在其所描述的图像中（与本页的某些地方相反，像标题或正文）会促使大脑去理解文字和图像之间的关系，并且能触发更多的神经元（Neurons to fire）。更多的神经元被触发等同于给大脑更多的机会，将该内容视为值得关注的事情，并且可能地把它记下来。

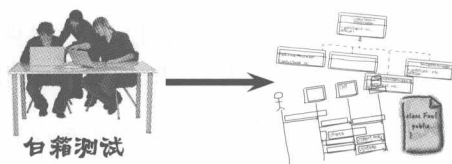
对话式的风格也相当有帮助，因为当人们意识到自己处在交谈之中时，他们会更加集中注意力，因为他们期望能一直跟得上谈话的内容至到对话结束。奇怪的是，你的大脑并不在乎你与书本之间的对话！另外一方面，如果你的写作风格既正式又枯燥，你的大脑会感到看书就像在被动地听一场报告一样，无须保持清醒。

但是，图像与对话式的风格，只不过是一个开端……



我们的做法

我们采用**图像**，因为你的大脑会调整你的注意力，让你注意到视觉效果，而不是文本文字。只要你的大脑被吸引，就一图值“千”字。并且，当文本和图像放在一起时，我们把文字嵌入到图像中，因为当文字在其所涉及的图像中时（而不是在图像说明或者埋藏在文字内某处），你的大脑会工作得更为有效。



我们采用**重复** (Redundancy)，以不同的表现形式、不同的媒介、多重的感知，说明同样一件事情，以增加学习内容烙印在你的大脑中多个区域的机会。

我们采用**意想不同** (Unexpected) 的方式，使用概念和图像，让你的大脑觉得新奇和有趣，并且我们采用多少具有一点**情感性**内容的图像和思想，让你的大脑感同身受。让你有感觉的事物，自然比较容易被记住，即使感受到的只不过是一点幽默、**新奇**或**有趣**。

我们采用**拟人化**，**对话式风格** (Conversational style)，因为当你的大脑相信你正处在对话之中，而不是被动地去听演讲之时，开发循环2便会付出更多的关注。即使当你是在阅读时（对话的对象是一本书），你的大脑还会这么做。



我们包含了80多项**活动**，因为当你在**做事情**时，而不是在读东西时，你的大脑会学会的更多，记忆的更多。并且，我们让练习活动保持在具有挑战性 (Challenging-yet-do-able)，又不会太难的程度，因为大多数人都喜欢这样的挑战。

我们采用**多重学习方式** (Multiple learning style)，因为你可能喜欢一步一步地学习过程，而其它人喜欢先了解整个轮廓，另外一些人则喜欢去直接看例子。但是，不论你自己是偏好于什么样的学习方式，每一个人都能从同一内容的多重表达方式中获益。



我们还考虑到了你的**左右脑** (Both sides of your brain) 运用，因为你的大脑参与的越多，就越有可能学会和记忆更多的事情，而且，你的精力集中的时间就越长。因为使用一边大脑，往往意味着另一边大脑有机会休息，你便可以学习的时间更长并且更有效率。



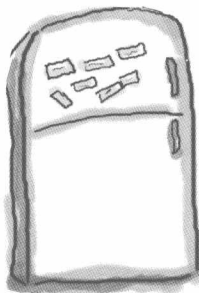
软件开发
工具箱

而且，我们也采用了**故事** (Story) 和练习，并用**多种角度** (more than one point of view) 来呈现，因为，当大脑被迫去做评估和判断时，会学习得更加深入。

在本书中，也包含了很多**挑战** (Challenges) 和练习，并采用问**问题** (Questions) 的方式。但是，对这些问题都没有直接给出答案，因为当你的大脑深入涉及某个问题时，你会学得更多、记得更牢。想想看——如果你只是在健身房中看别人锻炼的话，你如何能让自己达到塑身的效果。但是，我们做了充分的准备，确保当你的努力总是用在正确的事情上。你不会**花费额外的脑力**，去处理难于理解的例子，或是难以解析、行话充斥或咬文嚼字的论述。

我们采用**人物** (People)，在故事、例子和图像等等中，处处是人物。这是因为你也是一个人，你的大脑对人比对事物更加地注意。





把它剪下来，贴
在你的冰箱上。

让大脑顺从你的方法

那么，该做的我们都做了。剩下的就全靠你了。这些技巧是一个开头；倾听你的大脑的声音，弄清楚哪些对你有效、哪些对你无效。尝试尝试吧！

- ① 慢慢来，你理解的越多，需要记忆的就越少。
不要只是阅读，要停下来并且思考思考。当书上向你提问时，你不要跳过去直接找答案。设想一下，如果真的有人当面问你的问题，能迫使你的大脑思考得更深入，你就有机会学会并且记住更多的东西。
- ② 勤做练习，多记笔记。
在书中，我们安排了练习，但如果我们替你做练习，就像有人替你做训练一样。而且，不要只看练习，**使用铅笔作答**。大量事实证明，学习当中的体力活动能提高学习的效率。
- ③ 阅读“没有愚蠢的问题”的单元。
仔细阅读所有的“没有愚蠢的问题”的单元，它们不是无关紧要的内容——**它们是核心内容的一部分！**不要略过它们。
- ④ 把阅读本书作为上床前最后要阅读的内容，或至少是作为睡觉前的最后一件具有挑战性的事情。
学习的一部分反应发生在你放下书本之后，尤其把阅读到的知识转化为长期的记忆就更是如此。你的大脑需要自己的时间，去对阅读到的知识做更多的消化处理。如果在处理期间，你把新的知识塞进你的大脑，有些你刚学过的东西将会被遗忘。
- ⑤ 喝水，多喝水。
你的大脑需要浸泡在充沛的液体中，才能工作得最好。脱水（往往发生在你感到口渴前）减缓大脑的认知功能。
- ⑥ 谈论它，大声谈论它。
讲话激活了大脑的不同部位。如果你试图理解某些内容，或增加以后你记住它的可能性，就大声说出来。更好的办法是，尝试大声地解释给别人听。你会学习得更快，甚至触发许多新的想法，这是只通过阅读做不到的。
- ⑦ 倾听大脑的声音。
注意你的大脑是否在满负荷工作。如果你发现你自己开始蜻蜓点水，或者过目即忘，就是该休息的一会儿了。一旦你错过了某些重点，就应该放慢脚步，否则你将失去更多，甚至可能伤害学习的过程。
- ⑧ 感受事物。
你的大脑必须知道这件事情的重要性，你可以让自己融于到故事之中，为图片加上你自己的说明。即使抱怨笑话太出格，仍然比什么都没有感受到要好。
- ⑨ 编写大量的软件！
学好开发软件只有一种方式：要实际去开发软件（Actually develop software）。并且，这就是整本书所要你去做的事情。我们打算给你很多你要去捕捉的需求、去评估的技术，以及需要去测试和完善的代码：每一章都有练习题，每一道练习题都提出一个要你解决的问题。切勿跳过它们！当你解答这些练习题时，学习成效就会显现出来。每道练习题，我们都给出了**答案**，如果你真是卡住了，别不好意思去偷窥一下（人生难免遇到小波折）！但是，尽量在看答案之前，去解决所提出的问题。

读我

这是一段学习经历，而不是一本参考书。我们已刻意地剔除了所有可能妨碍学习的内容。并且，在你第一次阅读时，你必须要从首页开始，因为本书对读者已经阅读过哪些知识进行了假设。

我们假设你熟悉面向对象的编程。

教会你学习面向对象的编程可能要一整本书（如，《Head First OOA&D》）的篇幅，我们选择是本书的重点集中在软件开发的原理上，而不是软件设计或编程语言的基础知识。我们选择Java作为我们的例子，因为Java语言的应用已相当的普遍，并且相当能够自我说明（Self-documenting）；但是，本书中所讨论的每一件事情都应该是具有普适性的，适用于你正在使用的Java，C#，C++，或Visual Basic（或Ruby等等）各种语言。然而，如果你没有使用面向对象的语言编程的经验，在阅读部分软件代码时，你会碰到一定的困难。在这种情况下，我们极力推荐在你阅读本书的后面的章节之前，先熟悉一下其中的一种语言。

我们没有涵盖所有软件开发的过程。

有关软件代码编写的方法论方面的资料可谓是五花八门，我们并未试图去涵盖所有可能的开发代码的方法。恰恰相反，我们把重点集中在一些行之有效并能生产伟大的软件的技术之上。在第十二章中，我们特别讨论到一些调整流程的方法，让你的流程能够符合你的项目的实际。

活动不是可做可不做的。

练习题和活动不是额外附加的点缀，而是本书核心内容的一部分。其中，有一些能帮助记忆，有一些能帮助理解，还有一些能帮助你应用所学习到的知识。有些练习题只是让你思考如何解决这一问题，**不要跳过这些练习**。填字游戏是唯一不一定要去做的事情，但它们非常好地提供给你的大脑一个机会，在不同的段落中，让你回顾学过的专业词汇和技术术语。对这一部分，中文版将保留原文，方便读者玩此游戏。**重复是**