

/THEORY/IN/PRACTICE

# 软件开发路线图：

## 从学徒到高手

Apprenticeship Patterns:  
Guidance for the Aspiring  
Software Craftsman

REILLY®

机械工业出版社  
China Machine Press



HZ BOOKS  
华章科技

Dave H. Hoover & Adewale Oshineye 著

Ward Cunningham 序

王江平 译

# 软件开发路线： 从学徒到高手



David H. Hoover & Adewale Oshineye 著

Ward Cunningham 序

王江平 译

**O'REILLY®**

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo*

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社

## 图书在版编目 (CIP) 数据

软件开发者路线图：从学徒到高手/ (美) 胡佛 (Hoover, D. H.) 等著；王江平译. —北京：机械工业出版社，2010.7

书名原文：Apprenticeship Patterns: Guidance for the Aspiring Software Craftsman  
ISBN 978-7-111-31006-8

I. 软… II. ①胡… ②王… III. 软件开发 IV. TP311.52

中国版本图书馆CIP数据核字 (2010) 第113005号

北京市版权局著作权合同登记

图字：01-2009-7020号

©2009 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2010.  
Authorized translation of the English edition, 2009 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2009。

简体中文版由机械工业出版社出版 2010。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。

封底无防伪标均为盗版

本书法律顾问

北京市展达律师事务所

书 名/ 软件开发者路线图：从学徒到高手

书 号/ ISBN 978-7-111-31006-8

责任编辑/ 迟振春

封面设计/ Mark Paglietti, 张健

出版发行/ 机械工业出版社

地 址/ 北京市西城区百万庄大街22号 (邮政编码100037)

印 刷/ 北京京师印务有限公司

开 本/ 170毫米×242毫米 16开本 12.5印张

版 次/ 2010年8月第1版 第1次印刷

定 价/ 35.00元 (册)

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991; 88361066

购书热线：(010) 68326294; 88379649; 68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

## O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc.授权机械工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc.是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的*The Whole Internet User' Guide & Catalog*（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

## 作者简介

**Dave Hoover:** Obtiva首席技师，喜欢在开发软件的同时培养软件开发者，他的专长是向企业家们交付项目。

**Adewale Oshineye:** 软件工程师，从事过包括电子零售商销售网点系统、投资银行交易系统在内的各种大型项目开发。

# 目录

序 .....	1
前言 .....	5
软件工艺宣言 .....	19
<b>第1章 绪论</b> .....	<b>21</b>
什么是软件技能 .....	25
学徒期是什么 .....	31
学徒模式是什么 .....	32
模式来自哪里 .....	33
下一步做什么 .....	33
<b>第2章 空杯心态</b> .....	<b>35</b>
入门语言 .....	38
白色腰带 .....	45
释放激情 .....	49
具体技能 .....	51
暴露无知 .....	54
正视无知 .....	57
深水区域 .....	59

以退为进 .....	62
总结 .....	64
<b>第3章 走过漫漫长路 .....</b>	<b>67</b>
漫漫长路 .....	69
技重于艺 .....	71
持续动力 .....	74
培养激情 .....	77
自定路线 .....	80
使用头衔 .....	84
坚守阵地 .....	85
另辟蹊径 .....	87
总结 .....	89
<b>第4章 准确的自我评估 .....</b>	<b>93</b>
只求最差 .....	94
找人指导 .....	98
同道中人 .....	101
密切交往 .....	104
打扫地面 .....	107
总结 .....	109
<b>第5章 恒久学习 .....</b>	<b>113</b>
提高带宽 .....	114
不断实践 .....	118
质脆玩具 .....	121
使用源码 .....	124
且行且思 .....	128
记录所学 .....	131
分享所学 .....	133
建立环路 .....	136

学会失败 .....	139
总结 .....	140
<b>第6章 安排你的课程 .....</b>	<b>143</b>
阅读列表 .....	144
坚持阅读 .....	147
钻研名著 .....	148
深入挖掘 .....	150
常用工具 .....	155
总结 .....	158
<b>第7章 结束语 .....</b>	<b>161</b>
<b>附录A 模式列表 .....</b>	<b>167</b>
<b>附录B 一次学徒培训的号召 .....</b>	<b>171</b>
<b>附录C 回顾Obtiva学徒训练项目的第一年 .....</b>	<b>175</b>
<b>附录D 在线资源 .....</b>	<b>179</b>
<b>参考文献 .....</b>	<b>181</b>



# 序

25年前，我和Kent Beck坐在Tektronix技术中心的自助餐厅里，考虑着我们对Smalltalk-80的深入接触会给世界带来怎样的影响。

我对Kent说：真的不必担心。如果我们能做任何事情，我们会做什么呢？

“我想改变人们看待程序设计的方式。” Kent说。我表示赞同。那时，我们都认为行业发展进入了错误的方向，而我们都想逆转它。而且，令人惊奇的是，我们做到了。

我那时在自助餐厅中使用的这项技巧——“真的不必担心”这部分——是最早从我大学的指导老师那里学到的一种模式。他把该模式用在我身上，正如我把它用在Kent身上。这种行为我现在把它归结为一种模式，它使我和Kent敢于想象更远的目标，若非如此，那些目标看起来显得盲目。而一旦想到了，我们的目标就更容易实现了。

我把这种思考的技巧称为模式，因为它解决了我们经常遇到的一个问题：我们潜意识里会压抑自己的雄心壮志。本书中全是类似这样的技巧，它们用于解决各种不同的问题。我们说：模式解决问题。“真的不必担心”帮我和Kent解决了一个问题。它使我们真正去考虑自己本来就有的更大的想法，而且使我们克服了习惯性的自我压抑。

或许你也尝试过“真的不必担心”这一模式。如果没有，不妨试一试。



最强大的模式是那些可以反复运用并取得成功的模式。模式并不一定要新颖才有用。事实上，不新的模式才更好。另外，仅仅知道几个成型的模式名字也没有太大帮助。确定出一种模式，以后你每次谈论它的时候就不必重述整个故事。

快速翻阅这本书，你将看到很多模式，其中许多都会让你觉得熟悉。对任何一个，你可能都会说“我已知道这种模式了”——可能事实真的如此。然而，即使相关解决方案是常识性的，书中的模式仍然能提供两种帮助你方法。

第一，书中的模式更加完整。它们已经被研究、定性、归类并解释过。每一种模式都会带给你意外的收获。尽情享受这种收获吧，它会帮助你让已知的模式变得更加强大。

第二，这里的模式都是相互关联的。每一种模式都通向其他的多种模式。当你发现一种已经知晓的模式，沿着这些关联，你可以顺藤摸瓜找到其他一些尚未知晓或者从没想过可与之联合运用的模式。

我和Kent在Smalltalk-80中挖掘模式，我们找到了很多。我们将这些模式的概念讲给了同事们，并引发了一场小小的革命。我们改变了人们考虑程序设计的方式。从那时开始，人们写出了几十本有关模式以及如何使用它们的书籍。

我们的革命远未结束。慢慢地，模式术语越来越多，并成为敏捷（Agile）软件开发方法的基础。后来又出现了几十本书籍。

那现在为什么写这本书呢？好吧，我们的职业已经承载了过多的资源。关于我们的革命，有太多的信息可以获取，以至于没有人能完全吸收它们。然而，还是有些人努力做到了。他们消化了所有可以获得的建议，而且似乎总能在需要时信手拈来。他们是如何掌握到这种程度的呢？

本书的内容全都是教你如何掌握我们所处的复杂领域的模式。掌握不只是知晓。掌握是能帮你减轻负担的知晓。

举个例子，如果你记不住SUBSTR函数的参数顺序，可以到互联网上查一查。感谢上帝赐予我们互联网，它把我们的负担减轻了一点点。但是，当你运用本书中的模式，当你可以随时改善自己的工作，你会发现自己在写一种不同的代码，一种不依赖于了解SUBSTR参数顺序的代码。你将写出比SUBSTR飞得更高的程序，而这将大大减轻你的负担。

来自这次革命中的所有建议，除非能成为你的第二天性，否则就不会有太大的帮助。在软件领域，工艺的发展使我们认识到，让这些东西变成第二天性并不是我们的第二天性。这些模式是对这一过程的贡献，令人欢迎的贡献。

——Ward Cunningham





# 前言

不知而不知其不知者，愚者也——避之！  
不知而知其不知者，惑者也——授之！  
知之而不知其知之者，寐者也——醒之！  
知之而知其知之者，觉者也——从之！  
——Isabel Burton (1831—1896) 女士在  
《The Life of Captain Sir Richard F. Burton》  
(Richard F. Burton上尉的一生)  
一书中引用的阿拉伯谚语

## 本书目标

缺少经验的软件开发者常常面对进退两难的境地。为分享摆脱这类窘境的方法，我们写了这本书。我们指的不是技术上的窘境；本书中你不会找到任何Java设计模式或者Ruby on Rails诀窍。相反，我们所致力于解决的窘境是跟人有关的，是关于动机和士气的。作为专业软件开发领域的新人，你会面对一些艰难的决定，本书将帮助你做出这些决定。

## 读者对象

本书写给那些对软件开发有过一些体验并立志成为软件开发高手的人。你可能是一名Web应用开发者，或者是一名医疗设备程序员，或者正在为一家金融机构开发交易应用。你也可能刚从中学或大学毕业，并认为软件就是你的未来。



尽管本书是写给新人的，经验丰富的开发者仍可从本书内容中受益。有过几年开发经验的人，当从书中找到自己曾经面临过的窘境时，会不住地点头，而且还可能获得新的感悟，或至少获得一个新的词汇，用于描述他们自己使用或者建议给同行使用的方案。即使拥有十几年或者更长时间开发经验的人，特别是那些正在努力把握职业航向的人，也将会找到一些灵感和视角，用来抵御晋升到管理职位的诱惑。

## 写作过程

本书的构思始于2005年初，Stickyinds.com请Dave写关于软件工艺（Software Craftsmanship）的专栏。那时Dave认为自己是个（有经验的）学徒（apprentice），唯一让他写起来感觉舒服的主题就是学徒期（apprenticeship）。这使他开始考虑关于这个主题写些什么更好。大约就在那时，Dave读到一篇由软件开发者Chris Morris发表的网志<sup>注1</sup>，其中提到了吉他手Pat Metheny，而“只求最差”这一概念成了模式语言的种子。这粒种子很快从Dave的网志<sup>注2</sup>成长为Dave用来组织最初一些模式的个人wiki。最初的一些模式就是从Dave到那时（2000—2005）的职业生涯中提取出来的。

Dave知道，这些所谓的模式不能真正称为模式，除非它们确实是常见问题的一般解决办法。他开始通过三种方式从同行那里寻求反馈。第一，他开始在自己的网站上公开发表这些模式，通过公众点评的方式寻求反馈。第二，他开始访谈一些软件开发领域的思想领袖（主要通过电子邮件），并获得他们对最初几种模式的看法。第三，也是最重要的一种，Dave开始访谈一些经验较少的从业者，用他们新近的经验来检验这些模式。这些经验较少的从业者们还把一些Dave不曾遇到的或者在他的经验中不曾觉察到的模式介绍给他。就在这些关于学徒期的访问中Dave访问了Ade，经过双方同意，Ade加入到这个项目中并成为一名合著者。

---

注1：<http://clabs.org/blogki/index.cgi?page=/TheArts/BeTheWorst>.

注2：Red Squirrel Reflections（红松鼠沉思录），参见：<http://redsquirrel.com/cgi-bin/dave>。

我们（Dave和Ade）访谈了生活和工作在全世界很多地方的人们，从澳大利亚到印度再到瑞典。讨论的形式也五花八门，从LiveJournal网站的评论，到伦敦金融区中心一座美丽的被炸教堂遗址里进行的访问。

与此同时，其他一些人，如Laurent Bossavit、Daragh Farrell和Kraig Parkinson，他们勇敢地在各种练习班、研讨会和新手训练营中尝试使用这些模式。之后他们把收到的反馈传给了我们（Dave和Ade），而我们则竭尽全力地把这些反馈合并到我们的笔记中。

在2005年末的时候，我们在“程序的模式语言”（Pattern Languages of Programs, PLoP）研讨会<sup>注3</sup>上发起了一个主题小组。在PLoP上，我们得以将自己的工作呈现给那些经验丰富的模式作者（也称为牧羊人，shepherd）们，他们在模式的组织形式方面给了我们许多反馈，并用自己的程序开发经验测试了我们的论断。

大概在同一时间，来自O'Reilly Media的Mary Treseler联系我们，提出了出版这些模式的建议，并鼓励我们继续写作。她帮我们做了一些编辑工作，两年之后我们就书的出版达成协议。在那段时间里，通过电子邮件、用户组和讨论会，甚至午餐时的谈话，我们同数不清的同行就这些模式进行交流，同时我们继续在<http://apprenticeship.oreilly.com>在线社区上寻求反馈。

最终的结果就是你手上的这本书。这是一部基于与从业者之间的无数次交流的作品，也是一部对已有的关于学习、最佳效率心理学，以及所有我们能找到的与掌握知识相关的文献进行广泛研究的作品。随着阅读的深入，你将看到，除了通常的软件领域的名人之外，我们还会引用外科医生、舞蹈编导还有哲学家的话。我们相信，向任何一门学科的高手学习，我们都能收获很多。

---

注3：<http://hillside.net/plop/2005/group1.html>.



## 内容编排

模式是对特定上下文中某个问题的可重现解决方案的命名描述。这种描述应使读者对问题产生足够深入的理解，使他们或者能将所述方案应用到自己的情景中，或者能确定某个方案对他们的情景不适合。

本书由较长的几章组成，每章都填满了一组彼此相关的模式。模式的名字以首字母大写（这是指英文原版书——编辑注）拼出（如“质脆玩具”，Breakable Toys），而相关的模式则频繁引用。每一章都把其中的模式编排在一起，并提供一个对本章主题的介绍，以及一个进行总结的小节。本书的绪论部分为模式语言打下基础，而结束语部分则审视了有关技巧、学徒期和掌握专业知识的“大图景”。

## 模式的形式

我们的模式形式与众不同。如果你读过其他关于模式的书籍，你会看到在本书中我们尝试了一些不同的东西。同大多数模式语言相比，我们关于抽象外因和约束的章节不多，相关的讨论也较少。选择这种形式是基于来自审稿人和PLoP研讨会的广泛反馈。基于这些反馈，我们相信这种简单的结构将使我们的模式语言更易被目标读者接受和理解。

我们的模式都包含一个“情景分析”、一个“问题描述”和一种“解决方法”，然后是一组单一或多重的行动。“情景分析”设好基调，“问题描述”则确定模式所解决的问题。“解决方法”通常从问题的一句话分析开始，接着深入到与运用方法有关的各类细节问题上，其间描述该模式与其他模式的关系，还有支持该模式的故事和文献。

在每种模式接近尾声的地方都有一个“行动指南”的小节，这一小节描述了你可以立即付诸实践的具体事情，如果你愿意体验模式效果的话。这些行动可用作实施的示例，它们提供了一些练习，你可以立即投入实践，这样就不用担心模式是否可用于你身边的情形。

任何模式都应包含给定情景中一大类问题的一揽子解决方案，记住这一点很重要。模式是用来修正以适应具体情形的，而不是用来生搬硬套



的。因此，如果一种模式不能准确适用于你的情形，或者“行动指南”小节中任何一条看起来都不合适，那么你应该尝试沿着我们提供的材料进一步外推，看看能否构造出一些有用的东西。

大多数模式都以一个“参考模式”的小节结尾，这一小节指出了相关模式的页码。这样你阅读的时候就可以不采用线性顺序，而是采用一种迂回的顺序，从而加深你对不同模式间关系的理解。

## 用法

模式语言将创造的力量赋予每一个使用它的人，使他们创造出无数独特的新构造，正如一般意义上的“语言”给予他们创造无数句子的力量一样。

——《The Timeless Way of Building》（建筑的永恒之道），第167页

我们这个项目的目标是创建一种模式语言，来帮你定义自己的学徒过程。我们不可能知道你所处的实际环境，因此一定要考虑每种模式的情景分析和问题描述来确定它是否适合你。这些模式之间是相互联系的，可以联合使用来创造更强有力的体验。比如，尽管“找人指导”本身就是一种优秀的、经过时间检验的模式，但将它跟“密切交往”结合起来会更强大。另一方面，“暴露无知”更加依赖一些支持模式，比如“正视无知”和“以退为进”，而且需要多用一点技巧才能正确使用。跟所有的模式语言一样，你应该小心，不能过度使用这些模式。不要为使用各种模式而寻找借口，而应针对具体情形遴选最恰当的组合。

你不一定要按从头到尾的顺序阅读书中的模式。当初Dave阅读Christopher Alexander的《A Pattern Language》（模式语言）时，就是从书的中间开始，并沿着模式之间的关联来阅读，这带来了一种更有趣的学习体验。你也许只想简单浏览一下各种模式的情景分析和问题描述来找到跟你的实际情形最切合的模式。按这种方式来扫描模式可以让你在大脑中安装一些触发器来应对未来情形，那时，你会突然发现某些模式可以应用。

