

/THEORY/IN/PRACTICE

# The Productive Programmer

卓有成效的程序员 (影印版)

O'REILLY®

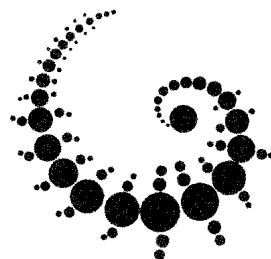
東南大學出版社

Neal Ford 著



卓有成效的程序员 (影印版)

# The Productive Programmer



Neal Ford  
foreword by David Bock

**O'REILLY®**

Beijing • Cambridge • Farnham • Koln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc 授权东南大学出版社出版

东南大学出版社

## 图书在版编目 (CIP) 数据

卓有成效的程序员: 英文 / (美) 福特 (Ford, N.)  
著. —影印本. —南京: 东南大学出版社, 2010.1

书名原文: The Productive Programmer

ISBN 978-7-5641-1924-9

I . 卓… II . 福… III . 程序设计—英文 IV . TP311.1

中国版本图书馆 CIP 数据核字 (2009) 第 205658 号

江苏省版权局著作权合同登记

图字: 10-2009-242 号

©2008 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2009. Authorized reprint of the original English edition, 2008 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 2008。

英文影印版由东南大学出版社出版 2009。此影印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。

## 卓有成效的程序员 (影印版)

---

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江 汉

网 址: <http://press.seu.edu.cn>

电子邮件: [press@seu.edu.cn](mailto:press@seu.edu.cn)

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 14 印张

字 数: 235 千字

版 次: 2010 年 1 月第 1 版

印 次: 2010 年 1 月第 1 次印刷

书 号: ISBN 978-7-5641-1924-9

印 数: 1~1500 册

定 价: 42.00 元 (册)

---

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

# FOREWORD

The individual productivity of programmers varies widely in our industry. What most of us might be able to get done in a week, some are able to get done in a day. Why is that? The short answer concerns mastery of the tools developers have at their disposal. The long answer is about the real *awareness* of the tools' capabilities and mastery of the thought process for using them. The truth lies somewhere between a methodology and a philosophy, and that is what Neal captures in this book.

The seeds of this book were planted in the fall of 2005, on a ride back to the airport. Neal asked me, "Do you think the world needs another book on regular expressions?" From there, the conversation turned to topics of books we wished existed. I thought back to a point in my career where I feel I made the leap from merely good to very productive, and how and why that happened. I said, "I don't know what the title of the book is, but the subtitle would be 'using the command line as an integrated development environment.'" At the time I credited my increased productivity to the acceleration I experienced using the bash shell, but it was more than that—it was my increasing familiarity with that tool as I stopped having to struggle to do things and could just get them done. We spent some time discussing that hyperproductivity and how to bottle it. Several years, untold conversations, and a series of lectures later, Neal has produced a definitive work on the subject.

In his book *Programming Perl* (O'Reilly), Larry Wall describes the three virtues of a programmer as "laziness, impatience, and hubris." Laziness, because you will expend effort to reduce the amount of overall work necessary. Impatience, because it will anger you if you are wasting time doing something the computer could do faster for you. And hubris, because excessive pride will make you write programs that other people won't say bad things about. This book doesn't use any of those words (and I used *grep* to check), but as you read on, you will find this sentiment echoed and expanded in this content.

There are several books that have had a great influence on my career, changing the way I see the world. I wish I had this book in hand 10 years ago; I'm sure it will have a profound influence on those who read it.

—David Bock  
Principal Consultant  
CodeSherpas



PREFACE

Many years ago, I taught training classes for experienced developers who were learning new technologies (like Java). The disparity between the productivity of the students always struck me: some were orders of magnitude more effective. And I don't mean in the tool they were using; I mean in their general interaction with the computer. I used to make a joke to a few of my colleagues that some of the people in the class weren't running their computers, they were walking them. Following a logical conclusion, that made me question my own productivity. Am I getting the most efficient use out of the computer I'm running (or walking)?

Fast-forward years later, and David Bock and I got into a conversation about this very thing. Many of our younger coworkers never really used command-line tools, and didn't understand how they could possibly offer more productivity than the elaborate IDEs of today. As David recounts in the foreword to this book, we chatted about this and decided to write a book about using the command line more effectively. We contacted a publisher, and started gathering all the command-line voodoo we could find from friends and coworkers.

Then, several things happened. David started his own consulting company, and he and his wife had their first children: triplets! Well, David now clearly has more on his hands than he can handle. At the same time, I was coming to the conclusion that a book purely about command-line tricks would be perhaps the most boring book ever written. At about that time, I was working on a project in Bangalore, and my pair-programmer partner, Mujir, was talking about code patterns and how to identify them. It hit me like a ton of bricks. I had been seeing patterns in all the recipes I'd been gathering. Instead of a massive collection of command-line tricks, the conversation should be about *identifying* what makes developers more productive. That's what you hold in your hands right now.

## Who This Book Is For

This isn't a book for end users who want to use their computers more effectively. It's a book about *programmer* productivity, which means I can make a lot of assumptions about the audience. Developers are the ultimate power users, so I don't spend a lot of time on basic stuff. A tech-savvy user should certainly learn something (especially in Part I), but the target remains developers.

There is no explicit order to this book, so feel free to wander around as you like or read it front to back. The only connections between the topics appear in unexpected ways, so reading it front to back may have a slight advantage, but not enough to suggest that's the only way to consume this book.

## Conventions Used in This Book

The following typographical conventions are used in this book:

### *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

### `Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

### **Constant width bold**

Shows commands or other text that should be typed literally by the user.

### *Constant width italic*

Shows text that should be replaced with user-supplied values or by values determined by context.

## Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*The Productive Programmer* by Neal Ford. Copyright 2008 Neal Ford, 978-0-596-51978-0."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).

## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (in the United States or Canada)  
707-829-0515 (international or local)  
707 829-0104 (fax)



We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

*<http://www.oreilly.com/catalog/9780596519780>*

To comment or ask technical questions about this book, send email to:

*[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)*

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

*<http://www.oreilly.com>*

## **Safari® Enabled**



When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *<http://safari.oreilly.com>*.

## **Acknowledgments**

This is the only part of the book my non-techy friends will read, so I'd better make it good. My entire life-support system has helped me greatly in this long, drawn-out book process. First, my family, especially my mom Hazel and dad Geary, but also my entire extended family, including my stepmother Sherrie and my stepdad Lloyd. The No Fluff, Just Stuff speakers, attendees, and the organizer Jay Zimmerman have helped me vet this material over many months, and the speakers in particular make the ridiculous amount of travel worthwhile. A special thanks goes to my ThoughtWorks colleagues: a group of people with whom I feel extraordinarily privileged to work. I've never before seen a company as committed to revolutionizing the way people write software, with such highly intelligent, passionate, dedicated, selfless people. I attribute at least some of this to the extraordinary Roy Singham, the founder of ThoughtWorks, and upon whom I have a bit of a man-crush, I think. Thanks to all my neighbors (both the non-garage and honorary garage ones), who don't know or care about any of this technology stuff, especially Kitty Lee, Diane and Jamie Coll, Betty Smith, and all the other current and former Executive Park neighbors (and yes that includes you Margie). Special thanks to my friends that now extend around the globe: Masoud Kamali, Frank Stepan, Sebastian Meyen, and the rest of the S&S crew. And, of course, the guys I see only in other countries, like Michael Li, and, even though they live only five miles away, Terry Dietzler and his wife Stacy, whose schedules far too rarely line up with mine. Thanks (even though they can't read this) to Isabella, Winston, and Parker, who don't care about technology but really

care about attention (on their terms, of course). A thanks to my friend Chuck, whose increasingly rare visits still manage to lighten my day. And, saving the most important for last, my wonderful wife Candy. All my speaker friends claim that she's a saint for allowing me to gallivant around the world, speaking about and writing software. She has graciously indulged my all-encompassing career because she knows I love it, but not as much as her. She's patiently waiting around until I retire or tire of all this, and I can spend all my time with her.

A special thanks goes out to the technical reviewers for this book. Without their hard work and dedication, this book would suffer lots of silly mistakes and confusing explanations. Thanks to Greg Ostravich (who has reviewed every book of mine for the last few years and gotten no recognition, unfortunately), Venkat Subramaniam, David Bock, Nathaniel Schutta, and Matthew McCullough.

## About the Author

---

**Neal Ford** is a software architect and Meme Wrangler at ThoughtWorks, a global IT consultancy with an exclusive focus on end-to-end software development and delivery. Before joining ThoughtWorks, Neal was the chief technology officer at The DSW Group, Ltd., a nationally recognized training and development firm. Neal has a degree in computer science, specializing in languages and compilers, from Georgia State University and a minor in mathematics, specializing in statistical analysis. He is also the designer and developer of applications, instructional materials, magazine articles, video presentations, and author of the books *Developing with Delphi: Object-Oriented Techniques* (Prentice-Hall), *JBuilder 3 Unleashed* (Sams), and *Art of Java Web Development* (Manning). He was editor of and contributor to the 2006 and 2007 editions of the *No Fluff, Just Stuff Anthology* (Pragmatic Bookshelf). His language proficiencies include Java, C#.NET, Ruby, Groovy, functional languages, Scheme, Object Pascal, C++, and C. His primary consulting focus is the design and construction of large-scale enterprise applications. Neal has taught on-site classes nationally and internationally to the military and to many Fortune 500 companies. He is also an internationally acclaimed speaker, having spoken at over 100 developer conferences worldwide, delivering more than 600 talks. If you have an insatiable curiosity about Neal, visit his web site at <http://www.nealford.com>. He welcomes feedback and can be reached at [nford@thoughtworks.com](mailto:nford@thoughtworks.com).

## Colophon

---

The cover image is a stock photograph from Corbis. The text font is Adobe's Meridien; the heading font is ITC Bailey.

## O'Reilly Media, Inc. 介绍

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，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 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。



# 出版说明

随着计算机技术的成熟和广泛应用,人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而,计算机领域的技术更新速度之快也是众所周知的,为了帮助国内技术人员在第一时间了解国外最新的技术,东南大学出版社和美国 O'Reilly Meida, Inc. 达成协议,将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作,以影印版或者简体中文版的形式呈献给读者。其中,影印版书籍力求与国外图书“同步”出版,并且“原汁原味”展现给读者。

我们真诚地希望,所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员 and 高校师生的学习和工作有所帮助,对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的影印版图书,包括:

- 《真实世界的 Haskell》(影印版)
- 《卓有成效的程序员》(影印版)
- 《Java Web 服务:建构与运行》(影印版)
- 《并行开发艺术》(影印版)
- 《使用 Perl 实现系统管理自动化 第二版》(影印版)
- 《Java 消息服务 第二版》(影印版)
- 《深入浅出网络管理》(影印版)
- 《Ruby 最佳实践》(影印版)
- 《更快速网站》(影印版)
- 《正则表达式 Cookbook》(影印版)
- 《flex 与 bison》(影印版)

# CONTENTS

	FOREWORD	vii
	PREFACE	ix
1	INTRODUCTION	1
	<i>Why a Book on Programmer Productivity?</i>	2
	<i>What This Book Is About</i>	3
	<i>Where to Go Now?</i>	5
<hr/>		
<b>Part One    MECHANICS</b>		
2	ACCELERATION	9
	<i>Launching Pad</i>	10
	<i>Accelerators</i>	18
	<i>Macros</i>	33
	<i>Summary</i>	35
3	FOCUS	37
	<i>Kill Distractions</i>	38
	<i>Search Trumps Navigation</i>	40
	<i>Find Hard Targets</i>	42
	<i>Use Rooted Views</i>	44
	<i>Use Sticky Attributes</i>	46
	<i>Use Project-Based Shortcuts</i>	47
	<i>Multiply Your Monitors</i>	48
	<i>Segregate Your Workspace with Virtual Desktops</i>	48
	<i>Summary</i>	50
4	AUTOMATION	51
	<i>Don't Reinvent Wheels</i>	53
	<i>Cache Stuff Locally</i>	53
	<i>Automate Your Interaction with Web Sites</i>	54
	<i>Interact with RSS Feeds</i>	54
	<i>Subvert Ant for Non-Build Tasks</i>	56
	<i>Subvert Rake for Common Tasks</i>	57
	<i>Subvert Selenium to Walk Web Pages</i>	58
	<i>Use Bash to Harvest Exception Counts</i>	60
	<i>Replace Batch Files with Windows Power Shell</i>	61
	<i>Use Mac OS X Automator to Delete Old Downloads</i>	62
	<i>Tame Command-Line Subversion</i>	62
	<i>Build a SQL Splitter in Ruby</i>	64
	<i>Justifying Automation</i>	65

	<i>Don't Shave Yaks</i>	67
	<i>Summary</i>	68
<b>5</b>	<b>CANONICALITY</b>	69
	<i>DRY Version Control</i>	70
	<i>Use a Canonical Build Machine</i>	72
	<i>Indirection</i>	73
	<i>Use Virtualization</i>	80
	<i>DRY Impedance Mismatches</i>	80
	<i>DRY Documentation</i>	88
	<i>Summary</i>	93
<hr/>		
<b>Part Two PRACTICE</b>		
<b>6</b>	<b>TEST-DRIVEN DESIGN</b>	97
	<i>Evolving Tests</i>	99
	<i>Code Coverage</i>	105
<b>7</b>	<b>STATIC ANALYSIS</b>	109
	<i>Byte Code Analysis</i>	110
	<i>Source Analysis</i>	112
	<i>Generate Metrics with Panopticode</i>	113
	<i>Analysis for Dynamic Languages</i>	116
<b>8</b>	<b>GOOD CITIZENSHIP</b>	119
	<i>Breaking Encapsulation</i>	120
	<i>Constructors</i>	121
	<i>Static Methods</i>	121
	<i>Criminal Behavior</i>	126
<b>9</b>	<b>YAGNI</b>	129
<b>10</b>	<b>ANCIENT PHILOSOPHERS</b>	135
	<i>Aristotle's Essential and Accidental Properties</i>	136
	<i>Occam's Razor</i>	137
	<i>The Law of Demeter</i>	140
	<i>Software Lore</i>	141
<b>11</b>	<b>QUESTION AUTHORITY</b>	143
	<i>Angry Monkeys</i>	144
	<i>Fluent Interfaces</i>	145
	<i>Anti-Objects</i>	147
<b>12</b>	<b>META-PROGRAMMING</b>	149
	<i>Java and Reflection</i>	150
	<i>Testing Java with Groovy</i>	151
	<i>Writing Fluent Interfaces</i>	152
	<i>Whither Meta-Programming?</i>	154
<b>13</b>	<b>COMPOSED METHOD AND SLAP</b>	155
	<i>Composed Method in Action</i>	156

	<i>SLAP</i>	160
<b>14</b>	<b>POLYGLOT PROGRAMMING</b>	165
	<i>How Did We Get Here? And Where Exactly Is Here?</i>	166
	<i>Where Are We Going? And How Do We Get There?</i>	169
	<i>Ola's Pyramid</i>	173
<b>15</b>	<b>FIND THE PERFECT TOOLS</b>	175
	<i>The Quest for the Perfect Editor</i>	176
	<i>The Candidates</i>	179
	<i>Choosing the Right Tool for the Job</i>	180
	<i>Un-Choosing the Wrong Tools</i>	186
<b>16</b>	<b>CONCLUSION: CARRYING ON THE CONVERSATION</b>	189
	<b>APPENDIX: BUILDING BLOCKS</b>	191
	<b>INDEX</b>	199





## CHAPTER ONE

### **Introduction**