

# 卓越程序员密码

## The Developer's Code

What Real Programmers Do

- 避免常犯的错误
- 养成优良的习惯
- 破解密码，你也可以卓越起来

[美] Ka Wai Cheung 著  
劳佳 译



人民邮电出版社  
POSTS & TELECOM PRESS

▲ 书

# 卓越程序员密码

The Developer's Code

What Real Programmers Do

[美] Ka Wai Cheung 著  
劳佳 译

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

卓越程序员密码 / (美) 张家为 (Cheung, K. W.) 著;  
劳佳译. — 北京: 人民邮电出版社, 2012. 11

(图灵程序设计丛书)

书名原文: The Developer's Code: What Real  
Programmers Do

ISBN 978-7-115-29508-8

I. ①卓… II. ①张… ②劳… III. ①程序设计—工  
程技术人员—工作方法 IV. ①TP311.1

中国版本图书馆CIP数据核字(2012)第232112号

## 内 容 提 要

本书集合了作者在软件行业里总结的第一手教训、体会和走过的弯路。话题涉及程序员生活的方方面面,例如,如何保持开发动力,如何提高生产力,如何摆脱软件的复杂性,如何与客户打交道,如何教导编程新手,何时进行自主开发,程序员的自豪感等。每个话题独立成篇、言简意赅,引人思考。

本书不仅适合编程老手阅读,也适合编程菜鸟学习,还适合想了解软件这个行业的人士阅读。

图灵程序设计丛书

## 卓越程序员密码

---

◆ 著 [美] Ka Wai Cheung

译 劳 佳

责任编辑 卢秀丽

执行编辑 岳新欣

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市海波印务有限公司印刷

◆ 开本: 880×1230 1/32

印张: 5.375

字数: 138千字

2012年11月第1版

印数: 1—4 000册

2012年11月河北第1次印刷

著作权合同登记号 图字: 01-2012-6909号

ISBN 978-7-115-29508-8

---

定价: 29.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

站在巨人的肩上  
**Standing on Shoulders of Giants**



[www.ituring.com.cn](http://www.ituring.com.cn)

# 译者序

这本小书名叫《卓越程序员密码》，但大部分内容讲的不是具体的技术。书中的话题涉及程序员生活的方方面面，每个话题独立成篇、言简意赅，读来多有切身之感，引人思考。譬如每日邮件繁忙，电话不断，如何保证工作效率？有些程序员朋友，自身水平很高，为何向别人讲解问题时却效果不佳？如何和不断提要求的客户周旋？作者在行业中浸淫多年，在团队建设、项目管理等方面都有独到的见解。当然，书中更少不了关于软件开发本身的技巧，复杂性管理、重构、代码生成等都是对实际工作很有指导意义的方法。

如果你从事开发，在上班的路上读上一两篇，也许书中的观点就能给一天的工作带来一点灵感。即使你从来不写程序，读读这本书也有助于了解一下这个热门而有趣的行业。要是你的男朋友刚好是个程序员，这本书或许能让你更好地走近他的生活，理解他的种种行为也说不定呢。

原文笔触轻快，语言流畅，读来丝毫没有读技术文献的枯燥感。作为译者，我也希望尽可能带给读者同样轻松愉快的阅读体验。限于水平和精力，译文不妥之处，还望读者指正。图灵岳新欣编辑和武卫东总编为译稿润色付出了辛勤的劳动，在此特别致谢。

最后值得一提的是，本书的翻译和编辑工作均在图灵社区上在线



## 2 | 译者序

完成。文本使用 Markdown 语言描述，实时更新，互动编辑，并逐章发布电子版。这种新的出版模式，消除了译者和编辑之间反复发稿的烦琐，也让读者能够随时追上最新的章节。在如今敏捷出版和移动阅读的大潮中，这一尝试显得尤为合宜，十分可喜。

劳佳

2012 年 8 月

# 中文版序

---

亲爱的读者：

在编程的世界里，我们会和各种各样的“语言”打交道。虽然我主要的服务器端开发语言是 C#，但我的工作方法却几乎可以完全应用到 Java、PHP、Ruby 或 Python 上。编程语言虽有不同，核心的编程思想、方法和架构却是高度类似的。我们只是用不同的方式来表达而已。

我们的工作方式也具有普遍性——如何保持干劲、提高成效，教学的重要性，如何与客户合作，如何发现合适的流程，勇于摆脱不好的流程。此外，我猜想，我通过编程学到的人生一课，不仅适合在美国工作的我们，也可能适用于在中国工作的你。

作为在美国出生的第一代华裔，我非常荣幸这本书能够被翻译成我父辈的语言。我在美国长大，在西方文化中生活，同时也有一些中国的传统价值观。我对于应该如何在这个行业工作的看法肯定受到了两种文化的影响。

无论你是刚刚投身编程事业，正在创业，还是在寻找一些新鲜的观点，我都希望这本书在现在和未来都能对你有所启发。

张家为

这本书的灵感大部分来自于我在 We Are Marmoth 的经验。这是

Dear readers,

In the world of programming, we work with “languages” of all different kinds. Though my primary server-side development language is C#, the way I work almost fully translates to the world of Java, PHP, Ruby, or Python. Core programming ideas, methodologies, and architectures are highly similar between these different coding worlds. We just express them in different ways.

The same is true for how we work - How we stay motivated and productive, the importance of teaching, working with clients, discovering the processes that work just-right and being brave enough to shed the ones that don't. And, I'd imagine that the life lessons I've learned through programming translate not just to how we work in the U.S., but how you might work in China.

As a first-generation American born to Chinese parents, I was tremendously humbled to have this book translated to my ancestral language. I grew up in the United States, living in a western culture while growing up with some traditional Chinese values. My opinions on how we ought to work in this industry have certainly been influenced by both cultures.

Whether your just beginning your career in programming, starting your own small business, or seeking some fresh perspectives, I do hope you find this book valuable now and in the future.

Sincerely, Ka Wai Cheung



# 致 谢

---

2010 年秋，这本书的初稿差不多已经写好了，于是我开始向几家技术出版社投稿。虽然有些反馈还不错，但一直没有人答应出版。多数出版社的意见主要有两条：一是这类书一般不好卖，二是我需要再加上更多的后续内容才值得出版。

Andy Hunt 和 Dave Thomas 看问题的角度与众不同。因此，首先我想感谢 Andy 和 Dave，他们和我一样，相信这本书在我们这个行业会有一席之地。本书能够忝列 Pragmatic Bookshelf 这套出色的丛书之中，我感到不胜荣幸。

像这样一本书迫切需要一位好编辑——能够客观地评价它，并在作者被盘根错节的细节羁绊的时候，能够从高处俯览全局。我要感谢 Brian P. Hogan 在整个过程中的出色工作。无论是从内容还是叙述方式而言，这本书胜过初稿之处都不可以道里计。

我要特别感谢 Andertoons.com 的 Mark Anderson。他富有创意的漫画和机智遍布全书，平添了几分幽默，让本书大为增色。

感谢 Derek Sivers、Bob Walsh、Caspar Dunant、Colin Yates、Juho Vepsäläinen、Steve Cholerton 和 Kim Shrier 抽出时间，热心仔细审读每一个章节。

这本书的灵感大部分来自于我在 We Are Mammoth 的经验，这是

## 2 致谢

我和 Craig Byrant 在 2006 年创立的 Web 开发工作室。在过去的五年里，我们不断挑战自己，思索我们对待工作的方式，并彼此坦陈我们的意见。感谢我的团队——Craig、Michael、Mustafa、Tom、Sam、Anthony、Jennifer、Grant 和 Lindsay，你们每天都带给我新的经验和教训。

# 目 录

|        |              |    |
|--------|--------------|----|
| 第 1 章  | 引言           | 1  |
| 1.1    | 谁是 21 世纪的程序员 | 2  |
| 1.2    | 吸取第一手教训      | 3  |
| 1.3    | 这本书写的是我们自己   | 4  |
| 第 2 章  | 比喻           | 5  |
| 第 1 篇  | 谨慎使用比喻       | 7  |
| 第 2 篇  | 规划完备，然后开工    | 8  |
| 第 3 篇  | 发行不过是第 1 版   | 10 |
| 第 4 篇  | “象牙塔”架构师的传说  | 12 |
| 第 5 篇  | 扔掉旧代码        | 15 |
| 第 6 篇  | 多元化胜于专业化     | 17 |
| 第 7 篇  | 比喻渐欲迷人眼      | 19 |
| 第 3 章  | 动力           | 21 |
| 第 8 篇  | 工作即福利        | 22 |
| 第 9 篇  | 从喜欢处入手       | 25 |
| 第 10 篇 | 莫求全          | 27 |
| 第 11 篇 | 休止一下         | 28 |
| 第 12 篇 | 早起先测试        | 30 |
| 第 13 篇 | 别在卧室里工作      | 32 |
| 第 14 篇 | 第一印象也就那么回事   | 34 |

## 2 | 目 录

|        |                     |     |
|--------|---------------------|-----|
| 第 15 篇 | 软件发行的情感价值 .....     | 38  |
| 第 16 篇 | 找个争论话题 .....        | 39  |
| 第 4 章  | 生产力 .....           | 41  |
| 第 17 篇 | 对休闲项目坚决说不 .....     | 42  |
| 第 18 篇 | 限制所有的因素 .....       | 46  |
| 第 19 篇 | 去掉时间表中的细节 .....     | 48  |
| 第 20 篇 | 每天改进产品的两个方面 .....   | 50  |
| 第 21 篇 | 为良好的工作环境投资 .....    | 52  |
| 第 22 篇 | 列一张个人待办事项清单 .....   | 56  |
| 第 23 篇 | 和团队一起安排免打扰时间 .....  | 62  |
| 第 24 篇 | 采用自治小团队的工作形式 .....  | 65  |
| 第 25 篇 | 提高生产力，避谈“我们” .....  | 67  |
| 第 5 章  | 复杂性 .....           | 71  |
| 第 26 篇 | “嗅”出坏的复杂性 .....     | 73  |
| 第 27 篇 | 关于“简单”的悖论 .....     | 75  |
| 第 28 篇 | 复杂性就像挑棍游戏 .....     | 78  |
| 第 29 篇 | 把复杂性藏起来 .....       | 79  |
| 第 30 篇 | “难编”可能意味着“难用” ..... | 81  |
| 第 31 篇 | 知道何时重构 .....        | 86  |
| 第 32 篇 | 确定编程的节奏 .....       | 92  |
| 第 6 章  | 教学 .....            | 94  |
| 第 33 篇 | 教学不同于编程 .....       | 95  |
| 第 34 篇 | 当心“知识魔咒” .....      | 97  |
| 第 35 篇 | 用浅显的例子 .....        | 99  |
| 第 36 篇 | 为简化不妨说谎 .....       | 102 |
| 第 37 篇 | 鼓励自主思维 .....        | 104 |
| 第 7 章  | 客户 .....            | 106 |
| 第 38 篇 | 刁钻的客户无处不在 .....     | 107 |
| 第 39 篇 | 软件黑魔法揭秘 .....       | 108 |
| 第 40 篇 | 设定软件的目标 .....       | 112 |

|              |                   |            |
|--------------|-------------------|------------|
| 第 41 篇       | 激发热情，相信自己 .....   | 113        |
| 第 42 篇       | 宽容大度，和蔼可亲 .....   | 115        |
| 第 43 篇       | 价值远不只是工时 .....    | 116        |
| 第 44 篇       | 尊重你的项目经理 .....    | 120        |
| <b>第 8 章</b> | <b>代码 .....</b>   | <b>123</b> |
| 第 45 篇       | 写代码是不得已而为之 .....  | 124        |
| 第 46 篇       | 拿来主义的文化 .....     | 126        |
| 第 47 篇       | 代码是最好的初级程序员 ..... | 130        |
| 第 48 篇       | 把机器和人的工作区分开 ..... | 135        |
| 第 49 篇       | 从核心开始生成代码 .....   | 140        |
| 第 50 篇       | 自主开发的情形 .....     | 147        |
| <b>第 9 章</b> | <b>自豪感 .....</b>  | <b>151</b> |
| 9.1          | 形象是个问题 .....      | 153        |
| 9.2          | 烹饪行业的一课 .....     | 154        |
| <b>参考文献</b>  | <b>.....</b>      | <b>159</b> |



## 第1章

# 引言

---

我被代码逼疯过——两次。

第一次，是我在大学一年级上的一门编程课。这是一门必修课，我就选了。结果这和我小时候看过无数次的电影场景完全是两码事。我并没有敲上几个简单命令，按下回车，然后就看到一个垃圾桶样的机器人和我说“你好”。

课堂上甚至连个垃圾桶机器人都没有，而全是一堆什么指针啊，内存分配啊，对象实例化之类的。我为了弄清楚这都是些什么玩意儿搞得焦头烂额。不过，有一件事是明摆着了——我根本不是编程的料。

我想成为一个艺术家，或者当个数学家。我想要有创造力还要能够精确——就像人们说的，左脑右脑一起上。编程似乎太偏向左半球了，可是我也想不出什么其他的职业道路，同时能够在两个世界施展。我迷茫了。

短短几年后，互联网的大潮彻底改变了编程的景象。编程突然就变得那么实际，那么触手可及，而且还和设计关系紧密。艺术和逻辑几乎被看做同样重要。第一次，我真的觉得能够享受这份工作，可以把我对于创造力和逻辑的热情都倾注到网络应用上。所以，虽然心怀抱志，我又回来编程了。

讲句老实话，我回来完全是另有原因。在之前的两年里，我学习

了其他很多学科，都似乎有太多太多的未解之谜。是去构思粒子物理里的大统一理论，还是去找最大的素数？毫无疑问，这些任务简直难到无法想象，让人望而生畏。我实在干不了这个。还有，那个讲存在主义的课也没能把事情说清楚。我是个年轻人，我想要的是答案，而不是越来越多的问题。

编程。我曾经唯恐避之不及的那个科目，现在成了我的避风港。不管怎么说，计算机科学还是人搞出来的。所有的事情肯定都有个答案。我期待的职业生涯，应该能让我这种寻求答案的人大展宏图——你能把甘醇的代码变成顾客、同事和客户常在的笑脸。规则都已经定好了，我们只要做就是了。困难无非就是在代码里面而已，我当时是这么想的。

我第二次来做编程，才发现又上当了——显然，实际远不是这么回事。

### 1.1 谁是 21 世纪的程序员

就像我在接下来的十五年中发现的那样，隐士不适合搞编程。什么有个超级技术狂，躲在幽暗的地下室里，挥汗如雨几个月，然后拿出最终产品，石破天惊，尽享荣华。绝对不是这么回事。

现在，应用程序是主流。即我们为每一位用户做的那种。客户可能了解我们如何工作，也可能完全没什么概念。我们的交付周期有时候是按照白驹过隙的星期来算的，而不是按照月或者年算的。人可能突然一下子就觉得疲惫不堪，而拖延可能是最容易走的那条路。对于我们——当今的开发人员来说，软件生产中所涉及的障碍已经远远超出了开发环境。

我有个好朋友经常和我开玩笑。“你工作的时候到底都干了些

啥？”她知道我是个程序员，但并不是很确定程序员具体都做些什么。她问我问题时的那种刻薄质问的口气，就像《上班一条虫》<sup>①</sup>里的那个办公室顾问 Bob Slydell 一样。

我是这么和她说的：我是一个无认证但是超级有逻辑的心理学家、理疗师、机械工、外交官、商人和教师，我所在的行业，迄今仍然每时每刻都有着新的诠释。

这就是我对当今的程序员能够给出的最简洁的定义了。

## 1.2 吸取第一手教训

我叫张家为。我是个程序员、设计师，还是芝加哥 We Are Mammoth<sup>②</sup>（我们是猛犸）公司的创始合伙人。我们为各行各业的客户制作应用软件，也制作我们自有的一些基于网页的软件。后面会再谈到这一点。

这本书集合了我自己在行业里总结的第一手的教训、体会和走过的弯路。对于编程老手来说，你可能会发现我写的某些轶事自己也经历过。我们可以一起大笑、欢呼或是流泪。对于刚刚开始征程的菜鸟来说，希望这本书能够为你进入行业的头几年助上一臂之力。

过去十五年中，我得到的教训数不胜数。下面仅举几个这本书里谈到的话题。

- ❑ 为什么软件行业中许多传统的开发流程和职责分配已经过时了，以及如何找出这些问题。
- ❑ 为什么要对消闲项目说“不”，为什么开放的时间表对生产力

① 《上班一条虫》(Office Space) 是 1999 年出品的一部美国电影。——译者注

② 我们的网站是 <http://www.wearemammoth.com>，我们的博客网址是 <http://blog.wearemammoth.com>。

至关重要。

- ❑ 协作的工作环境如何大幅提高我们的效率，又如何大大降低我们的效率。
- ❑ 如何让代码生成成为开发流程中自然而然的一部分，以及它除了让生成代码更快还有什么好处。
- ❑ 如何更好地与不能面对面交流的客户打交道，以及如何面对不管软件做了什么新改动都会大发雷霆的客户。
- ❑ 为什么大幅加薪和“员工是我们最宝贵的财富”这句古老的口号并不意味着更好的技术工作。
- ❑ 如何认识到软件本身已经太过复杂了。
- ❑ 如何成为更好的老师，让我们能够把知识在程序员中一代代传下去。

## 1.3 这本书写的是我们自己

这本书是写给各种各样的程序员的。不过，它和代码本身关系不大。不管你用的是 C#、Ruby、Python、PHP、Java、JavaScript 抑或是 ActionScript，都没什么要紧的。也无论你是否在搞数据库，写服务器端代码，还是做界面脚本，都无所谓。这本书讲的是在标识符和对象这个层面之外，专业程序员的方方面面。

这倒也不是说我们就把编程弃之一旁，会有些地方谈到代码。不过，在谈到代码的时候，我们会用一种不那么技术化的全局眼光来看。我们不会一下子给你列出一堆什么最佳实践或者设计模式之类的。这些东西有很多别的书都讲得很好了，我们只会适时提到一些。

这本书讲的是真正的、现代的程序员是怎样让我们的行业欣欣向荣的。我们开始吧。