

The Psychology of Computer Programming

温伯格技术思想三部曲

程序开发心理学

(银周年纪念版)

[美] Gerald M. Weinberg 著

邓俊辉 译

(Silver Anniversary Edition)



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

The Psychology of Computer Programming

温伯格技术思想三部曲

程序开发心理学

(银周年纪念版)

[美] Gerald M. Weinberg 著

邓俊辉 译



软件从业者思想启蒙的巨著

技术人生的必读经典

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书出自软件领域著名思想家,美国计算机名人堂首批成员之一的 Gerald M.Weinberg 温伯格之手。本书关注的是程序开发过程中人的因素,堪称这一领域的开山之作。本书的初稿完成于整整 40 年前,作者当时慧眼独具,前瞻性地提出了将程序开发作为一种人类行为来考察的观点。几十年来,本书已让无数读者从中受益,时至今日仍然是畅销全球的经典之作。

适合阅读本书的,不仅仅是处在软件行业第一线的程序员;各个软件开发单位的团队领导、项目主管、高层管理人员,以及人力资源经理、文档撰写人员、程序开发工具的设计者、程序开发语言的设计者,甚至所有其工作与程序开发有关的人,都能从本书中得到启发。

Original edition copyright ©1998,1971 by Gerald M. Weinberg. All rights reserved. Translation published by arrangement with Dorset House Publishing Co.,Inc.(www.dorsethouse.com) through the Chinese Connection Agency, a division of The Yao Enterprises, LLC.

本书中文简体版专有出版权由 The Yao Enterprises, LLC.代理 Dorset House Publishing Co.,Inc.授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字:01-2013-4963

图书在版编目(CIP)数据

程序开发心理学:银周年纪念版 / (美)温伯格 (Weinberg,G.M.) 著;邓俊辉译. —北京:电子工业出版社,2015.7

(温伯格技术思想三部曲)

书名原文: The psychology of computer programming

ISBN 978-7-121-25858-9

I. ①程… II. ①温… ②邓… III. ①软件开发—应用心理学 IV. ①TP311.52

中国版本图书馆 CIP 数据核字(2015)第 074360 号

策划编辑:刘 皎

责任编辑:徐津平

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:720×1000 1/16 印张:22.75 字数:387 千字

版 次:2015 年 7 月第 1 版

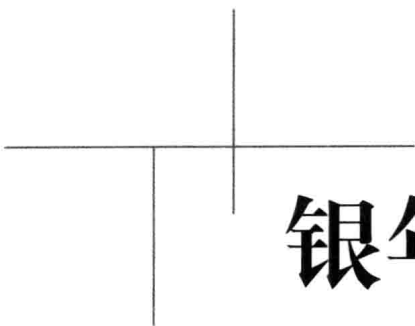
印 次:2015 年 7 月第 1 次印刷

定 价:79.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。



银周年纪念版序言

《程序开发心理学》是一本令人称奇的书——其令你感到奇特的程度，不亚于你听说有人能够活到 125 岁时的感受。我们所从事的这一行发展得飞快，据我所知，还没有其他任何一本计算机方面的图书，在初次出版之后，能够在长达 25 年的岁月中一直保持活力——而且这种活力到今天仍在继续。

令我自己更为惊讶的是，在写作本书时，我根本没想到它会有什么与众不同之处。在那之前的 15 年内，我一直在编写代码、带领程序员小组，以及为程序员提供培训和咨询服务。我确实曾经考虑过写一部关于计算机程序开发（在那时这项工作还鲜为外人所知）的小说，不过我意识到自己当时的写作技巧实属贫乏，写出来也不见得有人会信。我于 1969 年赴意大利休假八周，其间灵感突发，于是写成了《程序开发心理学》的初稿。

那时，我已经出版过几本关于如何为不同计算机开发程序的书，非常畅销，但是这本关于心理学的书对我来说却是一次新的探险。为了这本书的出版，我经历了许多挫折——有些是我在此前出版书籍时所不曾经历的。截至当时，McGraw-Hill 一直是我的出版商。这家出版社请了几位审阅人，他们看过我的手稿之后一致认定，没有人会愿意买这本书。于是我又转向 Prentice-Hall，他们勉强同意出版——但前提是，我必须同时把我的几本很赚钱的书也交由他们出版。我觉得还是找一家更为热情的出版社为好，因此又

将书稿同时交给四家出版社进行评阅。到这时，时间已经过去了两年，因此我决定哪家出版社首先表示愿意出版此书，我就交给它出版。最后四家出版社都表示愿意，但是其中最早回复我的一家出版社是 Van Nostrand Reinhold——在 1971 年，这本书终于付梓出版了。具有讽刺意味的是，就在这本书出版的当天，负责此书的编辑被解雇了，理由是他“对计算机出版一无所知”。

无论 Van Nostrand Reinhold 出版社对这位编辑如何评价，这本书很快就在技术类书籍中成为佼佼者，先后重印了 20 次，并且在 25 年之后仍在不断印刷。最后 Van Nostrand Reinhold 出版社将本书及他们所有的计算机图书的版权都转让给了另外一家大出版社——在此后的数月内，这本书就因为印刷中断而脱销。经过了几乎是遥无尽头的艰难谈判，我终于重新获得了这本书的版权，并委托 Dorset House 出版社出版本书的银周年纪念版^①。

我之所以要出版这本银周年纪念版，是基于以下几方面的考虑。

1. 让软件行业的新一代人士可以读到本书的原版。
2. 这一领域依然很年轻，我希望能够从历史发展的角度，为这一领域提供一些其所急需的纵览性观点。
3. 利用这个千载难逢的机会，对软件产业的发展做一评述，并对我以前的思想做一反思。

在这个新版本中，我并不想在软件心理学方面进行更新。首先，Ben 等人已经在这方面颇有建树，其水平远在鄙人之上。其次，正如 Ben 曾经评价的，实际上这本书谈论得更多的并不是软件心理学，而是软件人类学——在后续的几本书中，我在这两个方面都做了展开论述。

在重新审视本书时我的感受颇多，我认为至关重要的一点是，我希望能够暂时驻足片刻，以体会一下在过去的 25 年间，我自己及软件产业究竟前进了多远。对于软件产业而言，本书的第一版也许不能称作一个转折点，但对我自己而言，它的确是一个转折点。从那时起，我编写的代码少了很多，而且也更少参与对程序员团队的管理工作。然而另一方面，我却为成千上万的程序员及其团队的主管做了培训，并且为成百上千的软件项

^① Silver Anniversary Edition，银周年纪念版，即 25 周年纪念版。——译者注

目提供了咨询。我开始把更多的精力用于审查代码、设计方法、审查设计方案、了解需求及汇总需求。与此同时，我还专门花了很多时间，去培训那些有可能成为软件经理的人，并为他们提供咨询。有时我也确实会想回到从前的生活方式，那种整天唯一的工作就是编写代码的生活方式非常简单，不像现在这样需要兼顾这么多的其他事务。在软件产业界中，这是一种非常普遍的感受，尽管如此，却很少会有人按照自己的真实意愿身体力行。

如今，我可以明确看出的一点是，这本书的第一版实际上就是一张关于我自己所从事的研究方向（比如团队、领导艺术及方法、问题求解及问题定义等）的领航图。我把此后 20 多年的时光，都用于将这些大题目具体化，因为这些问题非常重要，却没有很好地为人们所理解。纵览自己在本书之后出版的一系列其他书籍，我觉得自己的确在致力于填补这些“空洞”。请允许我在此按照出版时间列出这些书的名字。

1973: *Structured Programming in PL/C, An Abecedarian*^①。针对程序开发技术及编写程序开发文本的新式教学方法，我们进行过一项实验。这本书中汇集了这项实验的成果。

1975: *An Introduction to General Systems Thinking*^②。这本书直接考察了关于系统思想的各种深入思考过程。（首印 20 多年之后，这本书仍在重印。）

1975: *Structured Programming*^③。Dennis Geller、Tom Plum 和我一起写成的这本书，荣获了结构化程序设计系列丛书奖。这本书中再次就程序开发思想及其教学的新方法进行了探讨。

1976: *High-Level COBOL Programming*^④。写作这本书的目的，在于帮助 COBOL 的程序员转变思维方式，但这一目标似乎并未实现。

① Gerald M. Weinberg, Norie Yasukawa Freedman, and Robert Marcus, *Structured Programming in PL/C: An Abecedarian* (New York: John Wiley & Sons, 1973).

② Gerald M. Weinberg, *An Introduction to General Systems Thinking* (New York: Wiley – Interscience, 1975).

③ Dennis P. Geller, Naomi Kleid, Thomas W-S. Plum, and Gerald M. Weinberg, *Structured Programming: A Film Series and Workbook* (Kansas City, Mo.: Edutronics Systems, 1975).

④ Gerald M. Weinberg et al., *High-Level COBOL Programming* (Cambridge, Mass.: Winthrop, 1977).

1977: *The Ethnotechnical Review Handbook*. Daniel Freedman 和我自掏腰包，出版了这本手册的第一版。写作这本书的目的，是希望激励人们，使他们在程序开发的每个阶段都去阅读程序，并且使他们学会如何阅读程序。在当时的情况下，面向市场的出版商中没有哪家愿意花钱出版这种技术评述性的图书。（这本书的最新版仍然在重印发行。^①）

1977: *Humanized Input: Techniques for Reliable Keyed Input*^②。这本书是在人性化界面设计方面的一次早期尝试。

1979: *The Ethnotechnical Review Handbook*。这本书后来做了重新修订，增添了我们后来在阅读及分析程序方面所学到的无数内容。

1979: *The Principles of Specification Design: Film Series and Workbook*^③。这本书是与 Bob Marcus 合作完成的，它代表了我们在改进问题定义方面的首次突破性成果。

1979: *On the Design of Stable Systems*。这本书是与我的妻子 Dani Weinberg 一起完成的。它将思维模式进行外推，以确定为了生存下去系统应该如何设计。（这本书的第二版仍在印刷发行，书名改为 *General Principles of Systems Design*^④。）

1982: *Are Your Lights On? How to Figure Out What the Problem Really Is*^⑤。Don Gause 和我合作完成此书，我们在此书中针对问题定义进行了深入研究，时至今日它依然是关于问题求解的一本畅销书。

① 这本书最终定稿书名为：Daniel P. Freedman and Gerald M. Weinberg, *Handbook of Walkthroughs, Inspections, and Technical Reviews*, 3rd ed. (New York: Dorset House Publishing, 1990).

② Tom Glib and Gerald M. Weinberg, *Humanized Input: Techniques for Reliable Keyed Input* (Cambridge, Mass.: Winthrop, 1977).

③ Robert Marcus and Gerald M. Weinberg, *The Principles of Specification Design: Film Series and Workbook* (Bowie, Md.: Brady Publishers, 1979).

④ 本书当前的版本是：Gerald M. Weinberg and Daniela Weinberg, *General Principles of Systems Design* (New York: Dorset House Publishing, 1988).

⑤ 本书当前的版本是：Donald C. Gause and Gerald M. Weinberg, *Are Your Lights On? How to Figure Out What the Problem Really Is* (New York: Dorset House Publishing, 1990).

1982: *Rethinking Systems Analysis & Design*^①。这本书的着眼点在于系统分析员的思考方法,或者说他们应该如何思考。

1982: *Understanding the Professional Programmer*^②。这本书的着眼点在于程序员的思考方式,或者说他们应该如何思考。

1985: *Computer Information Systems: An Introduction to Data Processing*^③。在这本书中, Dennis Geller 和我一起针对教学方法及软件工作中的思维过程等问题做了进一步探讨。

1985: *The Secrets of Consulting*^④。这本书针对软件开发人员(以及其他人员)与其客户之间的关系进行了讨论。这本书的创新之处大放异彩,直到今天它依然十分畅销——这足以说明,在咨询工作中有某种具有普遍性的原则可循。

1986: *Becoming a Technical Leader*^⑤。这本书着重讨论了领导方法及团队等问题,至今仍然销路看好。

1989: *Exploring Requirements: Quality Before Design*^⑥。这本书由 Don Gause 和我合作完成,我们在此书中将问题定义的问题探究到更深的层次。

1991: *What Did You Say? The Art of Giving and Receiving Feedback*^⑦。这本书是与 Edie 和 Charlie Seashore 合作完成的。在信息往返传递时,需要人性化的交互,本书总结了我们在这方面已经掌握的知识。

① 本书当前的版本是: Gerald M. Weinberg, *Rethinking Systems Analysis & Design* (New York: Dorset House Publishing, 1988).

② 本书当前的版本是: Gerald M. Weinberg, *Understanding the Professional Programmer* (New York: Dorset House Publishing, 1988).

③ Gerald M. Weinberg and Dennis P. Geller, *Computer Information Systems: An Introduction to Data Processing* (Boston: Little, Brown, 1985).

④ Gerald M. Weinberg, *The Secrets of Consulting* (New York: Dorset House Publishing, 1985).

⑤ Gerald M. Weinberg, *Becoming a Technical Leader* (New York: Dorset House Publishing, 1986).

⑥ Donald C. Gause and Gerald M. Weinberg, *Exploring Requirements: Quality Before Design* (New York: Dorset House Publishing, 1989).

⑦ Charles N. Seashore, Edith Whitfield Seashore, and Gerald M. Weinberg, *What Did You Say? The Art of Giving and Receiving Feedback* (Columbia, Md.: Bingham House Books, 1991).

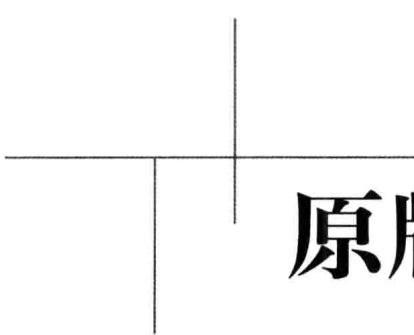
1991—1997: *Quality Software Management*^①。这是一套四卷的系列丛书。其中囊括了在软件成果及软件人员的管理方面本人所了解的所有内容。通过这一丛书，我也在系统的思想、测量、行为及变化等领域，从心理学的角度提出了新的课题。

直到完成了 *Quality Software Management* 系列丛书之后，我才感到自己在四分之一世纪之前开始着手研究的问题总算是有了一个总结。回顾这些年的发展过程，尤其是在重读本书的第一版时，我清楚地看到原来我是一个比自己预料中还要蹩脚的预言家，但是值得表扬的是，我已经料到自己是一个比预料的还要蹩脚的预言家。实际上，这个问题和所有的技术评论一样——虽然比自己预料的要差一些，但是比自己一直担心的情形还是要好一些。

在针对他人工作的技术评论中，我常给人家提一点忠告，对于本书的银周年纪念版，我也决定采纳自己提出的这个忠告，那就是：决不要掩盖自己的错误——因为，也许正是通过这些错误，我的读者才会获得更大的收获。为了在这方面对读者有所启发，我力求保持最初版本的固有风貌——不折不扣、货真价实的古董；同时，一旦灵魂受到震撼，我也会增加一些“事后诸葛亮式”的评注。我真诚地希望，你能够透过这个现代历史资料和文物的大杂烩获得某些启示，哪怕是能够对你有些许的帮助，也是鄙人莫大的荣幸与宽慰。

Gerald M. Weinberg

① Gerald M. Weinberg, *Quality Software Management, Vol. 1: Systems Thinking* (New York: Dorset House Publishing, 1992); *Quality Software Management, Vol. 2: First-Order Measurement* (New York: Dorset House Publishing, 1993); *Quality Software Management, Vol. 3: Congruent Action* (New York: Dorset House Publishing, 1994); *Quality Software Management, Vol. 4: Anticipating Change* (New York: Dorset House Publishing, 1997).



原版序言

写作本书的主要目的只有一个——启动一个新的研究领域：作为一种人类行为的程序开发，或者简而言之，程序开发心理学。所有的其他目标都从属于这个主要目标。例如，在力所能及的情况下，我会尽力增加本书的趣味性，同时淡化技术的味道，从而让更多人愿意阅读它：这些人不仅包括程序员，也包括程序开发的主管；同时，因为我们现在可以通过很多种方式参与到程序开发的过程中，所以还应包括与程序开发有关的其他所有人。我努力希望实现这样一个目标，即让读者在读完本书之后评价说：“嗯，程序开发的确不只是与软硬件有关。从今往后，我将从新的角度看待这些问题。”

正因为这是一个新的领域——或者说是对于那些已经为我们所熟知的事物的一种新的观察方法——所以，我们还不能为其中的某些思想找到“科学的”依据。本书中的许多观点，的确仅仅是作者的个人主张（而且是强烈的主张），然而除了本人多年以来的个人观察之外，这些主张尚没有更好的理论基础。毫无疑问，其中很多主张中存在的错误是显而易见的，即使是更有依据的一些思想，也同样可能存在谬误之处。但是我想指出的是，错误的思想与毫无思想之间存在天壤之别。如果哪位读者对其中的某个观点另有高见，我真诚地希望他能通过实验来证明我的谬误。

正如我所希望的，书中给出的大量实例说明，大量的误解与曲解阻碍了我们所从事的这个行业的发展。我和我的学生通过极其简单的实验，已经对此提出了挑战。然而与

问题本身的规模之广相比，我们所提供的素材实在是太有限了。各种不同的统计数据都显示出，今天担任程序员工作的人已经有数十万之众。如果我们的经验确实会对大家有所启发，如果程序员及其主管在读过本书之后，哪怕只是学会了将程序员当作活生生的人，而不是另外一台机器来看待，那么这些程序员就将更高效地发挥作用，同时会对自己的工作更加称心如意。

如果我们学会从心理学的角度看待问题，那么在硬件和软件的设计方面，我认为我们同样有可能取得巨大的进步。但愿设计师们在读过本书之后，能够学会运用心理学的原理指导其设计工作。当然，我并不指望区区一本书中的点滴思想和思索能够为他们直接提供所有的信息，但愿他们能够在本书的启发之下，学会从其他渠道获取信息。到目前为止，至少从工程或者数学的角度来看，程序开发仍然是一个深不可测的过程。然而实际上，这一过程与心理学有着血脉相连的关系，这种关系如此密切，以至于哪怕是在心理学方面的些许开窍，所带来的帮助都是难以估计的。这个断言并非无稽之谈——我个人及我的学生们在教学、学习及程序开发时都会将心理学牢记在心，我们的亲身经历与切身体会都证实了这一点。我希望本书的读者能够通过自己的工作实践，对这个断言进行验证。

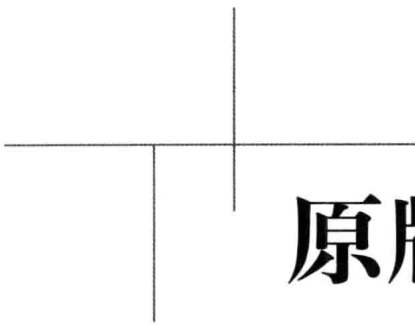
本书中显而易见的一个特点就是，其中的思想并非只来自我个人。实际上，这些思想都是在集成了很多人的经验和智慧之后才形成的——这些人中有很多是我的学员，他们来自 IBM 公司分别设在纽约、日内瓦，以及纽约州立大学宾厄姆顿分校的系统研究所。在这十年之间，我有幸能够在学员中结识如此之多的一线程序员；正因为如此，我才能够综合形形色色的程序开发环境的特点，将自己的一管之见不断扩充、完善——而要是凭借一个人的力量，恐怕任何人毕其一生也不可能会有如此丰富的经历。有一点大家应该都能理解，为了尊重（无论是清白的还是有过失的）当事人的隐私，我们将这些经历做了适当的加工整理。

如果没有与学生们长期的相互砥砺，没有他们丰富而且极具启发性的经历，这本书是不可能完成的。另一方面我也觉得，如果出于谦虚的考虑而毫无原则地将书中的一切谬误都归到自己的头上，那也是毫无意义之举。毫无疑问，因为这样或者那样的原因，的确会有一些与我相关的经历会被曲解。然而既然我已经采纳了这些经验，并且用来支持书中的某些观点，那么我就应该承担这部分责任——类似地，由于读者会根据自己的经

验与需求，对书中的所有观点进行权衡、取舍，所以读者应该对此自行负责。如果说有什么本来应该是我的最后一个愿望，那就是奢望书中提到的某些东西能够为读者接受，甚至成为他们的信条——然而，我并不希望读者在阅读过程中贸然接受某种观点，因为这种读书的态度正是我们应该努力戒除的。本书中提供的素材，只是读者在自己思想形成的过程中需要消化、吸收的食物，而不应该是个人思想的简单替代品。

我想在此感谢我的各位学生、老师及朋友，感谢他们对我个人的帮助，以及对本书的贡献。对我来说，老师、朋友和学生实际上是一个整体，并且没有区别，我也希望他们这样看待我。我要把这本书献给他们，我的这种感激之情不仅限于献辞那一页上，而是已融入于书的字里行间。我的妻子不仅在日常生活中给了我无微不至的关心，而且还从一位人类学家的视角，阅读了本书的每一页。我对妻子的感情，仅仅通过献辞是远不足以表达的。

Gerald M. Weinberg



原版序言评注

写作本书的主要目的只有一个——启动一个新的研究领域：作为一种人类行为的程序开发，或者简而言之，程序开发心理学。

四分之一世纪之前，序言的开头这样写道。现在有什么变化呢？我在年少轻狂时所定下的目标，是否已经达到？

唉，转眼间已经老了 25 岁，即使没有变得更聪明，至少我的锐气也应该有所收敛。现在确实有很多人已经开始把计算机程序开发当作一种人类行为进行研究，但是我却不敢妄言是因为在下的**这一拙作**，才开启了这个研究方向的大门——无论它是否真的可以被称为是一个“研究方向”。

关于本书用于教学和培训的建议

奥斯卡·王尔德^①曾经声称，无须在事先做任何准备，他都可以在任何集会场合，就任何话题侃侃而谈至少半个小时。当有一位存心刁难的听众请他就女王发表看法时，他依然桀骜不驯地答道：“但是先生，女王并不是一个话题呀。”

如果谈论的目标根本就称不上是一个话题，那么即使是像王尔德这样健谈的人，也无从谈起；但是许多人仍然不相信，程序开发心理学的确是一个话题。毫无疑问，很少有哪个大学开设这方面的课程，所以要是谁敢于第一个吃这个螃蟹，在大学中开辟这样一门新课程，其艰难之程度可想而知。像本书这样的一本教材，肯定会为这些人提供一些帮助，但是至少对于第一次尝试而言，就如何使用本书向他们提出一些善意的建议，应该不算过分。

本书是伴随着一门名为“程序开发心理学”的课程而逐步完成的。选修这门课的都是我们“计算机系统”项目中的研究生，他们都具有一定的程序开发经验，并且接受过至少一年的培训。对于那些尚不能独立编写程序、大学尚未毕业，或者虽然主修计算机系统或计算机专业，但是还没有达到本科四年级水平的人来说，选修这门课并非明智之举。

本课程是按照学术研讨会的形式进行的——因为在下自认为，如果学生已经懂得如何去阅读，课堂讲授完全就是多余的。本课程跨越两个学期，每学期各两个学分——这样安排的目的在于，可以使学生拥有尽可能多的时间，以积累个人经验。课程要求每位学生设计并完成一个小型实验，其中的一些实验对本书有过很大贡献（在书中可以读到有关的介绍）。选课的学生每星期开一次研讨会，每次两个多小时，事先他们需要阅读有关章节，同时根据对应的参考书目，就特定的讨论主题做好准备。

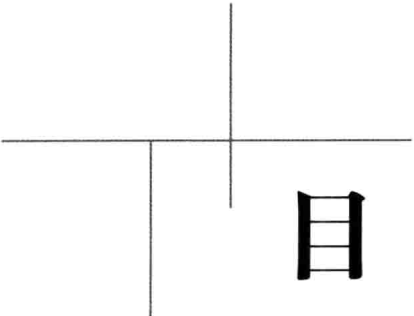
① Oscar Wilde, 奥斯卡·王尔德（1854—1900），19 世纪后半期欧洲文学史上的著名作家。1854 年 10 月 16 日，王尔德出生于爱尔兰的都柏林；1900 年 11 月 30 日，王尔德在加入天主教会第二天过世，并安葬于巴尼厄公墓；1909 年，王尔德的遗体移灵到拉雪兹神父国家公墓，长眠于此。19 世纪后半期的爱尔兰，依然还在英国的统治之下；1937 年，爱尔兰共和国成立；1949 年，英国终于承认爱尔兰独立。所以有人说王尔德是爱尔兰作家，也有人说王尔德是英国作家。无论如何，这位文学才子、诗人、戏剧家在文学上所取得的成就，不论是生前还是死后，都是举世公认的；同时，他的同性恋问题也引起了人们的很大争议。——译者注

关于课外阅读，我有一个很好的建议。由于有些参考资料属于内部出版，或者是不公开的技术报告，所以我们经常会事先为学生提供这些阅读资料的复印件。这种方式似乎可以使学生更有规律地进行课外阅读。

我们在进入有关“如何研究程序开发”这一主题时，有点急于求成，因为很少有学生会在行为科学方面有什么基础，更不用说行为科学方面的实验了。只有在这些方面做好准备之后，学生才具有着手进行实验的能力，同时我们才能够继续介绍后续的内容。是否要求学生进行实验，本课程没有统一的规定。尽管如此，倘若的确要求这样，那么很重要的一点就是：要让学生尽早动手。

如果不要求或者不希望学生进行实验，那么这门课就正好可以按照一个学期的时间来规划——既可以采用学术研讨会的形式，也可以采用课堂讲授的方式。无论采用何种方式，我都强烈建议在每个星期均花一段时间用于讨论。为了准备这些讨论，可以参考书中各章后面所附的思考题来确定讨论议题——可以要求学生做简单的笔记，也可以通过书面作业的形式。

对于暑期学院或者是管理培训研讨会等高强度的培训课程，由于其时间只有一到两周，所以这本书相应的用法也有所不同。在这种时间紧迫的情况下，我们只能忍痛割爱地取消实验。然而，还是可以通过一次简单的实验，让学生作为主体，切实体验一下实验的过程。本书中所建议的实验不胜枚举，其中任何一个都适宜于此类教学，但是如果时间限制得很严，就必须在事先做好最充分细致的准备。类似地，由于时间紧迫，必须保证每位学员都能够获得属于自己的一整套阅读材料。最后，为了缓冲学员的学习压力，可以将讨论分散开，安排到每次课堂讲授之后进行。如果学员已经是富有经验的程序员或程序开发主管，那么讨论就显得特别重要——因为他们在讨论过程中互相提供的信息，并不比从教员或者教材那里所获得的少，甚至可能会更多。对于他们来说，这些议题非常重要，而且非常贴近他们的实际工作，所以教员只需稍加点拨和组织，课程就将大获成功。



目 录

第一篇 作为人类行为的程序开发

1

阅读程序	4
一个例子	6
机器的局限	7
语言的局限	8
程序员的局限	9
历史遗留问题	10
规范	11
小结	12
思考题	12
参考资料	13
本章评注	14

2

优秀程序的要素	16
技术规范	18
日程计划	20
适应性	21
效率	23

小结	26
思考题	26
参考资料	27
本章评注	28

3

如何研究程序设计	30
自省	31
观察	33
实验	36
心理学测量	39
利用行为科学中的数据	42
小结	44
思考题	45
参考资料	46
本章评注	50

第二篇 作为社会行为的程序开发

4

程序开发组	59
正式与非正式组织机构	59
自然环境与社会结构	62
错误与唯我独尊	65
无私式程序开发	69
建立与维护程序开发的环境	73
小结	78
思考题	78
参考资料	80
本章评注	81