

纪念作者与计算机四十年的恋爱史

软件与系统思想家温伯格精粹译丛

质量· 软件·管理 (第II卷) ——阶测量

Quality Software Management: First-Order Measurement

[美] 杰拉尔德·温伯格 / 著

李先华 邢彦 张红艺 / 译



清华大学出版社



软件与系统思想家温伯格精粹译丛

质量· 软件·管理 (第II卷) 一阶测量

[美] 杰拉尔德·温伯格 / 著

李先华 邢彦 张红艺 / 译

清华大学出版社
北 京

Quality Software Management: First-Order Measurement

By Gerald M. Weinberg

EISBN: 0 932633 24 2

Copyright © 1993 by Dorset House Publishing Co., Inc. 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 Enterprise, LLC.

本书中文简体字版由 Dorset House Publishing Co., Inc. 授权清华大学出版社在中国境内独家出版、发行。

未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字: 01-2002-5678

版权所有,翻印必究。举报电话:010 62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

质量·软件·管理——一阶测量 (美)温伯格(Weinberg, G. M.)著;李先华,邢彦,张红艺译.

北京:清华大学出版社,2005.9

(软件与系统思想家温伯格精粹译丛)

书名原文:Quality Software Management: First-Order Measurement

ISBN 7-302-11730-6

I. 质… II. ①温…②李…③邢…④张… III. 软件质量—质量管理 IV. TP311.5

中国版本图书馆 CIP 数据核字(2005)第 101230 号

出版者:清华大学出版社

地址:北京清华大学学研大厦

<http://www.tup.com.cn>

邮编:100084

社总机:010-62770175

客户服务:010-62776969

责任编辑:贺岩

封面设计:李亚莉

版式设计:李尘工作室

印刷者:清华大学印刷厂

装订者:三河市李旗庄少明装订厂

发行者:新华书店总店北京发行所

开本:175×245 印张:24.75 插页:2 字数:307千字

版次:2005年9月第1版 2005年9月第1次印刷

书号:ISBN 7-302-11730-6/TP·7692

印数:1~5000

定价:43.00元

编委会名单

➔ **主 编** 张亚勤

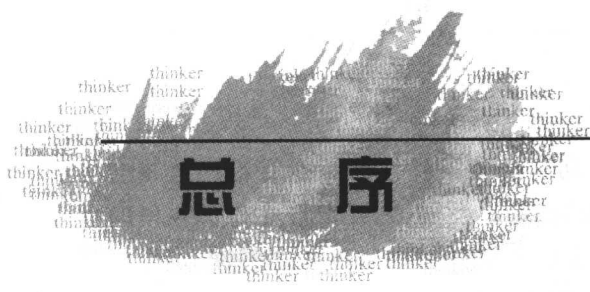
微软公司全球副总裁

➔ **顾 问** Gerald M. Weinberg

➔ **策 划** 熊妍妍

➔ **编委会委员** (以下按姓氏笔画排序)

- | | |
|-----|--------------------|
| 万起光 | 北京湘计立德信息技术有限公司 |
| 邓俊辉 | 清华大学计算机系计算机软件研究所 |
| 刘天北 | 北京奥捷特通信技术有限公司 |
| 朱于军 | 朗讯科技(中国)有限公司贝尔实验室 |
| 李 彤 | 北京诺亚舟管理咨询有限责任公司 |
| 杨作兴 | 北京方舟科技有限公司 |
| 周浩宇 | 创世嘉信企业管理顾问(北京)有限公司 |
| 邵维忠 | 北京大学软件学院 |
| 孟迎霞 | 《程序员》杂志社 |
| 章柏幸 | 北京津诚信业技术有限公司 |
| 胡庆培 | 新加坡国立大学计算机系 |
| 曾登高 | 中国软件网(CSDN) |
| 熊 节 | 《程序员》杂志社 |
| 潘加宇 | UMLChina |



总序

提到软件开发,人们就会想起微软。常有朋友问我,微软成功的秘诀是什么,怎样才能让软件走入千家万户。其实,这类问题早在 30 年前就有人完整地阐述和解答过;而且,即使是经历了这么长时间的技术革新,这些论述依然非常具有借鉴价值和启发性。解答问题的正是这一系列丛书的作者——尊敬的温伯格先生。

温伯格先生是从个体心理、组织行为和企业文化角度研究软件管理和软件工程的权威和代表人物,他有着程序员、系统设计师、咨询师、专业作家的多重身份。温伯格认为:软件的任务是为了解决某一个特定的问题,而软件开发者的任务却需要解决一系列的问题。他自称为“思考着的人”(thinker,而非人们为他定义的“思想家”),同时将他思考的结论和方法通过文字传递给数以百万计的读者。

温伯格还是一个实干家,他所创建的学校、培训基地,主持的大学、研讨会,给一代又一代软件工作者提供了“清新的空气”。温伯格最喜欢的一句话是中国的一句传统谚语:智者千虑,必有一失;愚者千虑,必有一得。思考是自作聪明者最大的弱项,也是成功者最大的财富。温伯格说,我们不能要求每个人都聪明异常,能够解决所有难题;但是我们必须持续思考,因为只有如此,我们才能明白自己在做什么。

“明白自己在做什么”,听起来是多么平常,具体操作起来却不是那么容易。我想,这可能也是温伯格花那么多心血在著书立说上的原因吧。十年树木,百年树人。温伯格著述颇丰,从早期集中在计算机和软件实务的作品,到 20 世纪 70 年代初《程序开发心理学》的出版,写作角度从软件开发实务过渡到更多的人文关怀。以他自己的话说,这些书的目的是为了探讨“人们是如何思考的”。尤其是在后续的《系统化思维导论》、《系统设计的一般

原理》和《你的灯亮着吗?》³本书中体现得尤为明显。

“明白自己在做什么”，是走向成功的必要条件。那些能够很早地领会或感悟到自然发展、社会发展、人类发展、行业发展、软件发展在很长一段时间内的可能趋势的先知先觉者，虽然在这个世界上不到万分之一，但是他们是时代的智者，只要他们愿意去做，他们就能够很快地获得成功。他们具有非常敏感的嗅觉和洞察力，能够很好地把握未来几年的软件需求，从而进行应用解决方案的设计、前卫体验理念的构建。或者说，他们能够在行业内把握方向，取得技术上的突破，特别是在一些尚未发掘的领域异军突起。他们属于时代或行业的领导者，其成功一半是天才，一半是勤奋。

还有一些人，他们对趋势的领会并不十分敏锐，但是他们最大的优点在于能够在经验的基础上踏实前进。他们的成功百分之九十九来自于学习和勤奋的实践。他们是时代和行业的中坚，是事实上的社会的缔造者，当然也是行业的建设者。他们能够清楚地知道自身的优势和劣势，根据时代和行业的现状，以及自身的经验和积累，进行主流软件开发、生产和实施。他们不一定掌握最新技术，但是他们一般来说资本和经验都非常充足，使他们保持中流砥柱位置的根本在于其能够正确认识到自身和外界的差距或互补，从而调整策略，后来居上或反败为胜。

“明白自己在做什么”，这种态度确保在进行软件开发和研究时保持理性和缜密的思考。经过了十多年的实践，温伯格先生称：“技术是毫无价值的。”我的理解是，如果我们都不知道自己的所作所为能给社会或自己带来什么，是根本无法找到那些有价值的技术的。而他所说无用的技术指的恰恰就是那些异想天开、不切合实际的无效劳动罢了。通过和温伯格先生的交谈和我自己在微软工作的经验，我可以负责地说，任何成功者都是其领域内的思考着的人，这种思考，使他们在不知不觉中逐渐向正确的方向转变；而温伯格的这一系列努力，正是让我们进行更深层次思考的提醒。

我相信，不论您是否从事软件开发、研究或管理工作，都能从温伯格先生谆谆的话语中受到启发。

微软公司全球副总裁 张亚勤

2003年8月



致中国读者

去年,我荣幸地得知,清华大学出版社将要引进出版拙著的一个中译本系列。作为作者,知道自己的作品将要结识成千上万的中国软件工程师、经理、测试员、咨询师以及其他相信技术能给我们带来更加美好的新世界的人们,我感到非常的惊喜。

在我早期的职业生涯中,我编写了大量计算机和软件方面的图书;但是,随着经验的增长,我发现如果我们在技术应用和构建之时对于其人文方面没有给予足够重视,技术就变得毫无价值——甚至是危险的了。于是,我决定在我的作品中加入人文领域的内容,并希望能够给读者带来这方面的思考和重视。

在这之后我出版的第一本书是《程序开发心理学》(*The Psychology of Computer Programming*)。这是一本研究软件开发、测试和维护中的关于人的过程的书。该书现在已经是银版了(自第一次出版至今已经有 25 年),这充分说明了人们对于理解其工作中人文部分的渴求。很高兴这本《程序开发心理学》也将成为这一中译本系列的其中一本。

清华大学出版社引进翻译我的系列作品,让我有机会将这些选本看作一个整体来思考,并且发现了其中的一些相通的主题。从我有记忆开始,我就对“人们如何思考”产生了浓厚的兴趣;当我还是一个小男孩时,世界上仅有的计算机常常被人称为“巨型大脑”。我当时就想,如果我搞清楚了这些巨型大脑的“思想”,我或许就可以更深入地了解人们是如何思考的。这就是我为什么一开始就做一个计算机程序员,而后又与计算机打了将近 50 年的交道;我学到了许多关于人们思考的知识,但是目前所知的还远远不够。

我的对于思考的兴趣在这些书中都有体现,而在以下 3 本中尤其清楚:即《系统化思维导论》(*An Introduction to General Systems Thinking*,这本书也已经

是银版了);它的姊妹篇《系统设计的一般原理》(*General Principles of Systems Design*,这本书是我与我的妻子 Dani 合著的,她是一位人类学者);一本《你的灯亮着吗?——发现问题的真正所在》(*Are Your Lights On? : How to Figure Out What the Problem Really Is*,这本书是我与 Don Gause 合著的);一本《探索需求设计前的质量》(*Exploring Requirements : Quality before Design*,这本书也是与 Don Gause 合著的),探讨的是人们如何思考他们在系统中的价值;还有一本《理解专业程序员》(*Understanding the Professional Programmer*),探讨的是程序员的独特思考过程。我对思考的兴趣很自然地延伸到如何帮助他人去清楚思考的方法上,于是我又写作了这个系列中的其他 3 本书:《咨询的奥秘——成功提出和获得建议的指南》(*The Secrets of Consulting : A Guide to Giving and Getting Advice Successfully*);《咨询的奥秘——咨询师的百宝箱》(*More Secrets of Consulting : The Consultant's Tool Kit*);《走查、审查与技术复审手册》(*The Handbook of Walkthroughs, Inspections, and Technical Reviews : Evaluating Programs, Projects, and Products*,这本书已经是第 3 版了)。

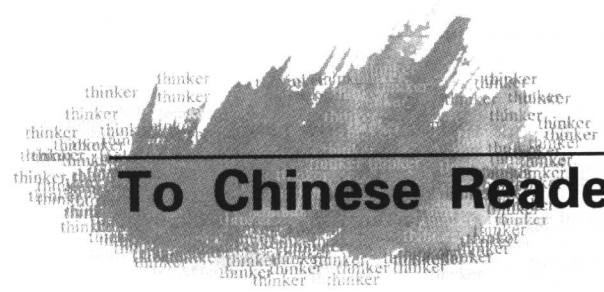
随着年龄的增长,我逐渐认识到清晰的思维并不是获取技术成功的充分条件,就算是思维最清楚的人也还需要一些道德和情感方面的领导才能。因此我又写了《成为技术领导者——解决问题的有机方法》(*Becoming a Technical Leader : An Organic Problem-Solving Approach*)。随后我又出版了 4 卷《质量·软件·管理》(*Quality Software Management*),其内容涵盖了系统思维、优先度量、协调行动和变更预期等,所有这些都是技术项目取得成功的关键。

在与各位译者的合作中,通过他们不同的文化视角和层面来审视我的作品,使我的思维和写作都获得了升华。我最大的希望就是这些译本同样也能够帮助你们——我的读者朋友——在你的项目,甚至是整个人生中获取更大的成功。最后,感谢你们的阅读。

杰拉尔德·温伯格

2002 年 10 月 11 日

Foreword by Gerald M. Weinberg



To Chinese Readers

Last year, I was honored to learn that Tsinghua University Press intended to publish a series of my books in Chinese translations. As an author, I'm thrilled to know that my work will now be within reach of thousands more software engineers, managers, testers, consultants, and other people concerned with using technology to build a new and better world.

Early in my career, I wrote numerous highly technical books on computers and software, but as I gained experience, I learned that technology is worthless—even dangerous—if we don't pay attention to the human aspects of both its use and its construction. I decided to add the human dimension to my work, and bring that dimension to the attention of my readers.

After making that decision, the first book I published was *The Psychology of Computer Programming*, a study of the human processes that enter into the development, testing, and maintenance of software. That book is now in its Silver Anniversary Edition (more than 25 years in print), testifying to the desire of people to understand that human dimension to their work. I was very pleased to learn that *The Psychology of Computer Programming* will be one of the books in this series of Chinese translations.

Having my books translated gives me an opportunity to reflect on them as a collection, and to perceive what themes they have in common. As long as I can recall, I was interested in how people think, and when I was a young boy, the few computers in the world were often referred to as “giant brains.” I thought that I might learn more about how people think by studying how these giant brains “thought”. That's how I first became a computer programmer, and after almost

50 years of working with computers, I've learned a lot about how people think—but I still have far more to learn than I already know.

My interest in thinking shows in all of these books, but is especially clear in *An Introduction to General Systems Thinking* (now also in a Silver Anniversary edition); in its companion volume, *General Principles of Systems Design* (written with my wife, Dani, who is an anthropologist); in *Are Your Lights On ? : How to Figure Out What the Problem Really Is* (written with Don Gause); in *Exploring Requirements : Quality before Design* (also written with Don, and is about how people think about what they really value in systems); and in *Understanding the Professional Programmer* (which is all about those thinking processes that are unique to those of us who are programmers). My interest naturally extended to methods of helping other people to think more clearly, which led me to write three other books in the series—*The Secrets of Consulting : A Guide to Giving and Getting Advice Successfully*; *More Secrets of Consulting : The Consultant's Tool Kit*; and the *Handbook of Walkthroughs, Inspections, and Technical Reviews : Evaluating Programs, Projects, and Products* (which is now in its third edition).

But as I grew older, I learned that clear thinking is not the only requirement for success in technology. Even the clearest thinkers require moral and emotional leadership, so I wrote *Becoming a Technical Leader : An Organic Problem-Solving Approach*, followed by my series of four *Quality Software Management* volumes. This series covers Systems Thinking, First-Order Measurement, Congruent Action, and Anticipating Change—all of which are essential for success in technical projects.

I have already begun to improve my own thinking and writing by working with the translators and seeing my work through different cultural eyes and brains. My fondest hope is that these translations will also help you, the reader, become more successful in your projects—and in your entire life. Thank you for reading them.

Gerald M. Weinberg

11 October, 2002

英文版序言

如果你能对自己谈论的对象进行测量,并用数字表达出来,那么你确实对它有所了解;然而如果你不能对它进行测量,也不能用数字把它表达出来,那么你的了解是肤浅的、是不能令人满意的……

——凯尔文(Kelvin)勋爵

经过 40 年的软件行业生涯我终于明白,为了在管理软件工程时能够高质量地工作,需要下列 3 个方面的基本能力:

1. 对复杂局面的洞察能力,这种能力使你首先能够对项目做好计划,然后对实施情况进行观察,并且采取必要的措施,保证项目按照计划进行,或者对计划进行修正。

2. 对正在发生的情形进行观察,并能够理解观察结果的意义,从而采取有效配合行为的能力。

3. 在各种困难的人际交往场合,即使你可能疑惑不解,可能生气,或者因为过于恐惧而想要转身逃走并躲藏起来,都能保持行为一致的能力。

以上 3 种能力对质量软件管理而言都是至关重要的,然而我却不想将本书写成气势磅礴的鸿篇巨著。因此,就像所有的软件质量经理一样,我将这一项目划分成 3 个子项目,每个子项目专门讨论这 3 种基本能力之一。第 I 卷《系统思维》(*Systems Thinking*)针对第一种能力——对复杂局面的洞察理解能力。第 III 卷《协调行动》(*Congruent Action*)针对的是第 3 种能力——即使是在情感十分强烈的情况下,依然能够采取正确行动的能力。在第 II 卷——《一阶测量》(*First-Order Measurement*)中,我敬请读者将注意力放在对正在发生的事情进行观察,并且能够理解观察结果的意义这一能力上。

我曾经为给本卷取一个恰当的书名而伤透了脑筋,最终才选择了一阶测量。首先,如果一本书出版时在书名中使用了“测量”一词,那么似乎作者

就不得不去引用 Kelvin 爵士的话。我一直都遵循这一传统,然而在他那句经常被引用的名言中,Kelvin 爵士的意思到底是什么,又不是什么?现在我感到必须对此作一分析。

首先,我必须指出,Kelvin 爵士是一位物理学家。因此当他说“如果你能对自己谈论的对象进行测量……那么你确实对它有所了解”的时候,他是从一位物理学家对自然界中具有普遍性法则的探寻的角度来说的。物理学家们进行测量,是为了在对事物的理解过程中获得乐趣;而被测量的究竟是什么,对他们来说并不太重要。

然而工程师们则正好相反,他们并不想了解普遍性法则,而只是与制造特定事物相关的特定法则。他们希望实现某种东西,因此任意的测量并不是他们需要了解的。

虽然从还是一个小男孩的时候起,我就一直希望能够与计算机共同工作,但是在我的学生年代里,学校没有任何计算机方面的培训。因此就像 Kelvin 爵士一样,我接受的也是物理学家式的培训。时至今日,我把自己当作一名软件工程师,尽管我依然热爱物理学,但我还是能够将下面的两类知识区分开来:

✓ 物理学方面的知识,是对自然界中普遍性法则的理解。

✓ 工程方面的知识,指的是懂得如何去做一件事。

支持去探求这两种理解的测量,也是不同的:我使用“三阶”测量一词,来表示那种支持物理学家对普遍性法则进行探求的测量。例如,对热力学第一定律的测试结果告诉我们,实际上,永动机是不可能制造出来的。

工程师们所感兴趣的,只是我称之为“一阶”和“二阶”测量,用来制造机器,以及随后进行调整。特别地,一阶测量只适用于制造某种物体的任务。通俗地说,这类测量就是所谓的“信封背面”(在英国,称之为“香烟盒背面”)的计算。^①这些测量支持的是粗率的计算(rule of thumb)、简略的草图(rough sketch),以及“草草做出的”(quick and dirty)或者“仅凭经验和直觉做出的”(seat of the pants)的估计。这些就是使我们能够首先制造出一台机器的那种测量。

二阶测量要更为精细一些。通过它们,可以对系统进行优化,使得成本

① Back of the envelop calculation,在信封背面上就可以完成的计算,指对数字进行简短的分析。——译者注

更低或运转更快。也可以对机器进行调整,使其性能达到最有效的状态。尽管二阶、三阶测量对软件工程专业的发展非常重要,但它们实际上并不涉及典型软件工程主管的日常问题。或许,下面这个故事可以解释我为什么会做出上述论断。

Tweedle 先生及夫人有一对 15 岁的双胞胎孩子,Dum 和 Dee。他们正在学习驾驶汽车。Dee 曾经驾驶过自家的轿车 10 次,她安全记录非常理想。虽然 Dum 也开过 10 次车,但是他却出过 3 次车祸。Tweedle 夫人对丈夫说,为了不至于弄出人命,他必须和 Dum 谈谈。

Tweedle 先生与 Dum 的谈话首先从对这 3 次事故的回顾入手,然后他问 Dum:“你自己觉得应该如何解释呢?”

“嗯,在 10 次旅行中出现 3 次事故,这个比例还不算糟糕。”

“我或许可以同意你的说法。”Tweedle 先生说,“但是,你的妹妹 Dee 也旅行过 10 次,她的挡风玻璃上连一点石头屑都没有。”

“你说得不错,”Dum 回答说,“但是我的单位里程耗油量要比她的少。而且,我也没有使轮胎上沾上泥巴。”

“噢,”Tweedle 先生说,“我还没有考虑到过这些问题。好吧,从今往后开车时还是要更加小心,同时在单位里程耗油量和汽车的干净程度等方面也要继续保持优点。现在我必须和你妹妹谈谈她的驾驶习惯。”

即使是我们当中那些曾经被自己十几岁的孩子蒙蔽过的人,也能看出,Tweedle 先生作为父母简直就像一位白痴。然而,在你如此苛刻地对他进行评判之前,请记住,这只不过是一则寓言式的故事。那么作为一则寓言,这个故事要告诉我们什么呢?从软件工程中,你也许同样可以发现 3/10 这个数字——这个数字,正是我们众多客户的那些大型软件项目中**最终不能完成的**比例。Capers Jones、Tom DeMarco 和 Tom Gilb 都是软件质量方面的专家,他们都向我证实:根据他们的经验,30%这一比例是一贯不变的。

设想如果你是一家企业的软件开发经理,而这家企业已经做过 10 个项目,其中有 3 项以失败告终。虽然这些项目都失败了,但是项目主管们却仍然能够给你展示某些精确的测量,这些测量表明:与成功的那 7 个项目相比,他们的“单位成本代码生成行数”更多,“每行代码平均错误数”更低。难道,你会因此转而去找负责其他几个项目的主管,与他们讨论一下他们的管理习惯?显然不会。

正如 John von Neumann 所言：“在你对自己正在谈论什么尚未搞清楚之前，追求精确是毫无意义的。”然而，当众多的管理者使自己的企业深陷于那些完全基于二阶测量的测试程序时，他们做的正是 Neumann 所说的那种情况。许多其他软件工程书籍的作者鼓励管理者的，正是这种不计后果的行为方式。这些书籍中标题含有“测量”或“度量”等关键词，主要讨论的都是二阶测量，有的时候甚至是三阶测量。软件测量方面的大师之一 Bill Silver 也承认：“这令人难过，但却是事实：大多数软件测试程序都不管用。”^[1]

Silver 的看法也得到了 Caper Jones 的确认，后者根据经验认为，80% 的软件测量程序在两年内就会消失。我自己的客户也承认了这种观点。他们的情况，与 Silver 指出的导致失败的首先、最重要的原因是吻合的：

“几乎没有哪家公司能拥有一种质量文化，为测量提供一种积极的氛围。”

请允许我更坦率地表述这一观点。如果一家公司在每 10 项主要的软件项目中，就有 3 项折戟沉沙，那么它就还谈不上二阶测量。而如果这样一家公司企图安装一套主要基于二阶测量方法的测量程序，那么这种做法，弊要大于利。并且很可能花费了上百万美元进行的测量，最后却搞得一团混乱。这并不是“测量的积极氛围”。

来自软件质量文化的数据显示：就目前而言，只有小部分组织拥有可以支持二阶测量程序的文化。^[2]我撰写《质量·软件·管理》系列丛书的目的，就是为了帮助这些组织的管理者，提高他们的管理质量，进而改进其组织。

本卷的目的，在于帮助这些组织的管理者，达到这样一种水平——即能够对二阶甚至三阶测量的使用有目的地加以考虑。如果你所在的组织通过努力能够按时在预算范围之内交付软件产品，而且这些产品在使用期中，能令客户愉悦——并且在至少 99% 的时间里，你都能够做到如此——那么你就没有必要阅读这卷《一阶测量》。实际上，如果你的组织能够符合这些标准，那么我会很高兴地将你为本卷支付的书款如数奉还，来交换你在质量管理方面获得的教训。

然而，要是你的组织还没有达到这些标准，那么我就希望能够告诉你，应该如何创建“一种测量的积极氛围”，来避免那些已经毁灭了如此之多测量程序的、代价高昂的错误；同时也告诉你，应该如何简单而有效地对事物进行测量——这些东西将会帮助你的组织，始终如一地生产出你想要的高质量软件。

致谢

通过审查、讨论、演示、实验和举例等形式，下列各位曾经为本书的改进作出重要的贡献，我想在此对他们表示感谢：

Jim Batterson

Naomi Karten

Jinny Batterson

Norm Kerth

Mike Dedolph

Brian Nejme

Peter de Jager

Lynne Nix

Tom DeMarco

Judy Noe

Kevin Fjeldsted

David Robinson

John Freer

Wayne Strider

Phil Fuhrer

Linda Swirczek

Dawn Guido

Dani Weinberg

Payson Hall

Ellie Williamson

Jim Highsmith

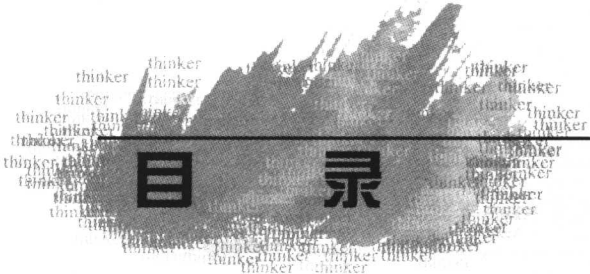
Janice Wormington

Capers Jones

Gus Zimmerman

我同样感谢 Change Artists 公司和其他几家我的客户组织，以及问题解决领导能力研究班、组织转变研讨会、质量软件管理研究班和其他教学实验的众多参与者。

在这份名单里，我本来还想加上许多经理和员工的名字，然而，尽管他们提供了生动的故事，他们却不希望透露自己的名字——当你读过这些故事以后，就会明白他们之所以希望如此的显而易见的原因。凡是为我提供了保密信息的个人和客户，我都在此将其名字隐去。



目 录

导言:观察的模型	(1)
----------------	-----

第 I 篇 摄 入

第 1 章 观察为什么重要	(7)
管理失败:危机还是错觉?	(7)
理解文化	(8)
运转中的文化观察模式	(12)
观察模式的效果比较	(21)
有益的提示与补充	(21)
提要	(22)
练习	(24)
第 2 章 选择观察对象	(29)
摄入阶段	(30)
选择观察哪些东西:关于 1 的寓言	(33)
有效观察模型的需求	(35)
管理者的错觉和渐进式衰退	(40)
有益的提示与补充	(41)
提要	(43)
练习	(45)
第 3 章 可视化产品	(47)
利用各种感觉道	(48)

使软件可见	(49)
使软件可观察	(55)
产品的开放性是驾驭自如式文化的关键	(58)
有益的提示与补充	(59)
提要	(61)
练习	(62)
第4章 可视化过程	(65)
过程开放性是未雨绸缪式文化的关键	(66)
识别未雨绸缪式组织	(68)
过程图的词汇	(69)
项目控制面板	(78)
有益的提示与补充	(79)
提要	(79)
练习	(80)

第Ⅱ篇 赋 意

第5章 关于解释的案例分析	(85)
偏移图:比较承诺日期和交付日期	(85)
对公司A的项目偏移图的解释	(88)
对公司B的项目偏移图的解释	(90)
公司C的文化	(96)
有益的提示与补充	(98)
提要	(99)
练习	(100)
第6章 从观察结果获取含义时的陷阱	(101)
3个解释的法则	(101)
运用数据询问	(102)
解释观察	(105)
在测量上的花费过多过快	(105)
陷阱	(110)