

B e c o m i n g   a   T e c h n i c a l   L e a d e r :

温伯格技术思想三部曲

# 成为技术领导者

掌握全面解决问题的方法

[美] Gerald M. Weinberg 著  
余晟 译

A n   O r g a n i c   P r o b l e m - S o l v i n g   A p p r o a c h

B e c o m i n g   a   T e c h n i c a l   L e a d e r :

温伯格技术思想三部曲

# 成为技术领导者

## 掌握全面解决问题的方法

[美] Gerald M. Weinberg 著  
余晟 译

A n   O r g a n i c   P r o b l e m - S o l v i n g   A p p r o a c h

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

搞定技术问题并不简单，但与人打交道也并非易事。作为一个技术专家，你是否在走上管理岗位时遇到了各种不适“症状”？本书将帮助你成为一个成功的解决问题的领导者。书中温伯格从一个反思者的角度阐述了要成为一个成功的解决问题的领导者必备的3个技能——MOI，即激励（Motivation）、组织（Organization）和创新（Innovation）。同时还提供了一些方法以供读者对上述3个技能进行自我分析和自我完善，最终帮助你走上成功的解决问题的领导者之路。不同于其他讲述管理的图书，温伯格并没有大量阐释原理、定义，而是通过现实当中的故事启发读者自己找到最终的答案。本书在美国项目管理学会的《项目管理知识体系指南》（PMBOK）中，被确定为重要的参考文献。

Original edition copyright ©1986 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：01-2013-4964

### 图书在版编目（CIP）数据

成为技术领导者：掌握全面解决问题的方法 /（美）温伯格（Weinberg, G.M.）著；余晟译. —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：Becoming a technical leader: an organic problem-solving approach

ISBN 978-7-121-25860-2

I. ①成… II. ①温… ②余… III. ①领导学 IV. ①C933

中国版本图书馆 CIP 数据核字(2015)第 074359 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：15 字数：255 千字

版 次：2015 年 7 月第 1 版

印 次：2015 年 7 月第 1 次印刷

定 价：69.00 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 zllts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。



# 再版译者序

## 我所理解的技术领导力

一晃六年，本书要再版重印了。回想刚刚开始翻译这本书时，我还忙碌在程序开发的一线，对领导技术团队并没有太多经验；如今，也能差强人意地带领技术团队支撑年销售额数亿元的业务。一路走来跌跌撞撞，所幸没有中途倒下。思考其中的原因，除去运气，除去身边同事朋友的支持，翻译《成为技术领导者：掌握全面解决问题的方法》也是不容忽视的因素。

很多人都知道，“职场童年”非常重要，一个人最初工作的几年，在什么样的环境里，得到过什么样的锻炼，很可能决定了他整个职业生涯的走向。同样的道理，“领导力童年”也很重要，一个人对领导力的最初接触和认知，也会深深影响他对于“领导”和“领导力”的观点，甚至领导作风。所以，在我还忙于一线开发的时候，通过翻译《成为技术领导者：掌握全面解决问题的方法》，“生吞活剥”了一整套关于领导力的学说，基本“塑造”了我关于领导力的认知，深深影响了我作为技术领导的管理风格和价值取向，因此也对很多问题有了自己的判断——前段时间和另一位掌管公司技术的朋友聊天，说起那种“执行力超至上”的领导风格，我们都认为，尽管或许能出结果，但不是好的领导风格。

怎样成为好的技术领导？《成为技术领导者：掌握全面解决问题的方法》的作者温伯格给出了一系列答案。以技术人员的身份，我觉得最受用的几点是：通过写日记来加深自

我认识；评估自己的能力应该用乘法而不是加法；从哪里寻找改进所需要的时间。

在生活中，我们大多会有一些自己的习惯和坚持，却习焉不察，从未想过它们是否有道理。然而，如果需要不断改进自己，就必然需要不断观察和审视关于自己的一切，包括习焉不察的坚持。对此，温伯格给出的建议是，每天抽一点时间来写日记，不需要抒情，不需要感慨，忠实记录自己就好。开始我对此也有些怀疑，觉得写日记不过是小朋友过家家。但真正坚持之后才发现收获很大。因为静下心来写日记让我发现，在生活里，自己真正在乎什么，不在乎什么。有一些东西我很在乎，但成长经历不同的人并不在乎，所以其他人并不是有意冒犯。还有一些东西我不在乎，却是某个人群特别在意的，考虑到其他人的感受，以后还是多加小心为好。

评估自己的能力“用乘法而不是加法”，可能初看起来不太容易理解，但也很好懂。人的能力通常是参差不齐的，有长处就必然有短处。要想成为好的技术人员，技术是不能“一俊遮百丑”的。以1分为满分，如果你的技术有0.8分，表达能力只有0.1分，总分可能不是 $(0.8+0.1)/2=0.45$ 分，而是 $0.8 \times 0.1=0.08$ 分。如果给你时间把自己的技能提升0.1分，总分可以到0.09分，而用来提高表达能力，则可以达到0.16分。哪种选择对个人更有利，其实是一目了然的。不幸的是，大家通常在潜意识里更愿意遮挡自己的短处，更习惯于训练提高自己的长处。程序员更是如此，面对非技术问题，他们往往更希望从技术方面解决。仅仅是因为他们更“喜欢”从这方面入手，而没有考虑这样做是否真的有效率。想要成为优秀的技术人员，一定要能克服情感的抵触，注重补齐短板。

我在抓虾网（早年一个很流行的在线RSS阅读网站）工作时曾有过深刻的体会。有一次我们的抓取调度出了问题，抓取新浪博客过于频繁，被新浪封锁了服务器的IP。当时大家想出的办法就是找一大堆代理服务器，通过代理来抓取。从技术人员的角度出发，这种思路是自然而然的。我当时觉得还有更好的解法，于是我尝试给新浪打电话说明情况，一下午的时间打了超过20个电话，终于找到了决定封锁IP的人。我向他说明情况，并说明我们已经修正了程序，不会再出现这种情况，他验证之后便解除了封锁。有了这段经历，不管是关于纯技术问题还是业务问题，我的思路都开阔了很多，也深刻明白了单纯靠技术是不能“一俊遮百丑”的。

关于改变自己所需要的时间，温伯格的一句话让我印象很深，“如果你想做某件事情却一直找不到时间，那多半是你其实不想做”。想要改变，尤其是自我改变，通常不会像

上级布置的任务那样，有明确的压力和期限，所以改变也停留在“想”而已。网络上经常可以看到类似的问题：道理我都懂，但就是行动不起来。所以很多人在纠结，希望有什么办法提高行动力。但是在我看来，要解决这个问题，第一步是承认自己其实不想实践这些道理。

如果确认自己想去做这件事情，又苦于找不到时间，温伯格给了三个建议：第一，对已经分配的任务，不要反复纠结；第二，对实现过程中的细节，不要反复纠结；第三，不要让自己的生活被层出不穷的危机所支配。比如对于“缺乏行动力”的问题，如果你真的希望提升行动力，应该首先制订计划，制订好计划之后应该按时推行，在这个过程中可以容忍错误和异常，但不要轻易纠结于计划本身。在实现过程中，不要过分纠结细节，比如学英语，捧着一本书刚看了个开头，就纠结是先学语法呢，还是先背单词，抑或开始纠结是这本书更好一点，还是那本书更好一点。更重要的是，要想有时间做自己的事情，应当把一切事情保持在“井然有序”的状态，哪怕平时需要花更多的时间来维护，这样才不会被各种意外所支配。我曾经见过很多程序员，每天忙于改正线上的各种问题甚至乐在其中，却从来不想怎么让程序保持在“自主稳定运行”的状态，还一个劲地抱怨“工作辛苦，生活忙碌”。也正是因为如此，我才大力提倡程序员要“横向发展”，要操心程序运行的整套环境，才能真正把自己解放出来。

当然，既然名为《成为技术领导者：掌握全面解决问题的方法》，温伯格讲得最多的还是“如何成为技术领导”。温伯格反复强调的根本观点是，人不应当被作为机器对待，尤其因为技术工作强调思考和创新，所以技术工作者更不应当被视为机器，而应当被视为种子——蕴含内在力量，会不断发展成熟的种子。所谓领导力，就是创造一个环境，让所有的人都可以发挥出比单干时更大的价值，并不断成长。以此为基础，领导力的培养和发挥，需要注意激励、组织和创新这三个方面。

关于领导的激励，已经有很多的论述。不幸，通常的激励似乎是从行为主义心理学的角度出发的，认为简单机械的“奖励/惩罚”就可以对员工起到引导和归束的作用。但这种理论其实是行不通的。赫兹博格的“激励-保健因子”理论指出，员工在不同的阶段所看重的方面是不同的，简单说员工刚开始更看重的是个人生活、工作环境、薪金福利等“基本因子”，满足之后则寻求学习与发展、工作乐趣、成就与肯定等“激励因子”，而简单的“奖励/惩罚”在这些方面并不能奏效。更重要的是，因为技术工作的核心之一是创新，简

单的“奖励/惩罚”并不能催生创新。按照我的经验，激励的作用更多是树立正确的价值观，这种价值观既符合公司的利益，又兼顾个人的成长，而且还要能落实到真实的工作中来。

在很多公司，有一些程序员是众望所归的“明星”：程序出了什么问题，找他们可以第一个响应，而且他们可以非常投入地解决问题，哪怕加班加点也无所谓。可是如果仔细思考，这样的程序员有不少就是麻烦的制造者，因为他们写不出高质量的程序，只能以“高度的责任心”在线上除错。而且，这样的程序员往往因为态度好、加班多，受到大量的关注和鼓励。还有一些程序员，他们或许沉默寡言，或许不爱加班，但他们总能提交高质量的程序，上线之后就不需要自己再操心。不幸的是，这样的程序员往往不会获得关注，颁发奖励的时候也论不上他们。

身为技术人员，很多人都知道这两种做法的优劣，但因为外界（领导）没有明确的褒贬，很多人并不敢坚持自己的选择。所以，如果希望成为好的技术领导，一定要注重激励的方面。在日常工作中，技术领导应当持续表扬和鼓励能提供高质量程序的行为（哪怕日常不怎么说话），而不是提交质量一般但努力除错的行为。有这种持续的激励，才有可能塑造正确的价值观，给有潜力但还在摇摆、困惑的成员发出清晰的信号，从而打造高质量、迅速成长的团队。

技术领导需要注意的第二个方面是组织。前段时间我和一位读过 MBA 的朋友聊天，他说很多领导对于招人的定义就是：因为我忙不过来，所以我需要一个人帮我做这个。他评价说：“其实，这类领导需要的不是员工，而是劳工”。用我的话说，这种组织不叫团队，只能叫团伙。

既然领导力的表现是创造让所有人都能成长，都能发挥更大价值的环境，当然不能把所有的人当成可以互相替换的棋子。按照温伯格的意见，好的组织应当是“全面的”（Organic，也可以翻译为“有机的”），也就是可以互相取长补短，形成一股合力。假设一个团队里没有产品经理，那么虽然客户对产品的要求并不是太高，程序员也有一定的产品意识，交付的软件堪称能用，但技术领导应当看到，关于产品的工作其实消耗了开发人员大量的时间，而且开发人员本身并不“愿意”从事产品方面的工作。所以应当考虑补充产品人员，并让产品和开发协调工作，形成  $1+1>2$  的结果，提升整个团队的效率。同样的道理，如果团队里多数开发人员都比较沉闷，在继续招聘开发人员的时候，就应当优先考虑开朗外向的

性格。

组织的全面，还体现在一个方面，即它是自组织的，各级的情况和任务可以在对应的级别自动自发地完成。或者用温伯格的话说：“在全面的组织中每个人都能解决问题，做出决策，执行这些决策。而领导不需要对各种问题亲自出面，亲自做决策，亲自执行”。这种观点也可以在其他相关书籍中得到验证，例如 Uncle Bob 就在《程序员的职业素养》中再三强调，团队要有凝聚力。要想打造全面的组织，有凝聚力的团队，温伯格列出了几种需要警惕的行为，包括“只抓大目标”（特别强调执行力的领导作风就是如此）、把人当成机器来看待（忽略人的情感和潜力）、事必躬亲（下属不应当仅仅是领导完成任务的手段）、奖励低效的组织（回到价值观的树立），等等。虽然我们日常工作中无法做到彻底戒除，但只有尽力避免这样的行为，才能真正营造全面的组织，形成有凝聚力的团队。

技术领导需要关心的第三个方面是创新。许多技术领导本身对技术非常有兴趣，所以他们自己在创新方面是没有问题的。但是身为领导，仅仅自己创新是不够的。既然相信人不是机器，既然相信软件开发是需要创造力的工作，那么就应当鼓励每个人的创新，为团队营造勇于创新的气氛。

余晟



# 译者序

## 身份证掉了怎么办？捡起来

大概是 1992 年左右，我第一次见到“脑筋急转弯”，马上就被这种新鲜玩意迷住了。可惜，当年那些稀奇古怪的问题，如今都忘得差不多了，记得的只有一道题——身份证掉了怎么办？捡起来。

记得这道题，并不是因为它很难，而是因为，这些年来的许多经历，事后细细咀嚼起来，答案就像“身份证掉了怎么办？捡起来”一样简单、直白，又荒诞、无奈——或者怨天尤人，或者凭空慨叹；可是，薄薄的一张身份证，就是没法捡起来。

2006 年末我着手翻译《精通正则表达式（第 3 版）》的经历，就是如此。

试译的时候，我一晚上就交出了六页译稿。接下来约定全书的交稿时间，我不假思索地说，最多六个月。可书拿到手的那一瞬间，整个人就傻了——五百页，整整五百页，以前是轻松畅快的阅读体验，现在是堆积如山的任务列表！

按计划，我每天应该翻译四到五页，然而我只坚持了两天就中止了——每天下班累得要死，还得做这劳什子翻译，五百页何时能结束啊？

于是干脆自我放纵，过了一周“轻松加愉快”的生活，到周末，我甚至在想，原来生活这么美妙，干脆取消合同，交违约金算了吧。

然而，就在这一瞬间，我忽然又想到了那个脑筋急转弯——“身份证掉了怎么办？捡起来”。现在的这张身份证，无非就是在六个月内完成这本书的翻译嘛。而捡起来的方式，就是坚持每天翻译而已。答案简单到完全不存在其他的可能，简单到毫无风险。

于是我开始忍住内心的厌倦，慢慢捡起这张掉落身份证。五页，十页；一章，两章……每做一点点，我都安慰自己说，对了，我在一点点地捡起那张身份证。

说来也奇怪，日子一天天地过去，我反倒习惯了这样的生活，厌倦的感觉也渐渐消失了。

全书翻译就要结束的时候，一次误操作弄丢了上万字的译稿，怎样也无法恢复。痛惜了半小时之后，我忽地又一次想到了“身份证掉了怎么办”这个问题。好吧，既然无法挽回，重新来过就是了。

这一回，为什么恢复得这么快？是习惯使然吗？

这个问题，一直保持到去年翻译温伯格的《成为技术领导者：掌握全面解决问题的方法》：在第4章，温伯格用现身说法讲解了，我们在成长过程中必然会遭遇“峡谷”，他总结说：

高原-峡谷模型描述了一个轮回周期，也描述了一个元周期——也就是周期的周期，这是个螺旋，就好像鹦鹉螺上一圈圈的纹路。走出低谷，不仅仅是登上另一个高原，同时也是在另一个高原上前进，这就是学习如何成长的高原。我的确是在学习新的语言，但更重要的是，我在学习“如何学习计算机语言”。这种元学习（meta-learning），清楚地说明了我对学习新语言的情绪。我不再焦虑，不再抵触，不再觉得毫无价值，而是感到激动，充满创造活力，能够处理几乎任何事情。

说到这里，我豁然开朗了：是的，现在我能更快地接受现实的解决办法，直接开始“捡身份证”，原因就在这里；我也发现，自己开始翻译《成为技术领导者：掌握全面解决问题的方法》的时候，已经没有了当初的煎熬：划分好进度，每日执行就是了。

《成为技术领导者：掌握全面解决问题的方法》一路翻译下来，我越来越赞同序言的说法：温伯格的这本书“卑之无甚高论”，并没有太多稀奇。如果“成为技术领导”是那

张身份证，温伯格的书就不是传说中的《九阴真经》、《葵花宝典》，他只是告诉你，怎样才能把它“捡”起来——你仍然需要弯腰，仍然需要伸手，而他所说的只是，如何弯腰更省力，怎样伸手更合理。这时候，更大的问题或许在于，你是否愿意（是否能）按部就班地去做。

譬如在第7章，温伯格介绍了一种加深自我认识的“不起眼”的办法，写日记：每天五分钟就可以，写下自己想写的内容，过一个月再回顾；第10章介绍了一种办法，照它一步步地分析，我们发现“小小的”误会背后，原来隐藏着长长的逻辑链条；第13章给出了协调“自己跟自己过不去”的那些“做人原则”与现实的途径……

而温伯格的解法也是按部就班：通过持续的日记加深自我观察；通过分析交流的每个环节理解沟通的误会；通过每次一点小的转变化解自己觉得“不可更改”的原则。这样的做法，很“蠢”，但很有效——这是我的现身说法。

类似的例子，在这本书中还有许多。读完这本书，你或许会明白，尽管我们奢望“取其精华、去其糟粕”，然而这世界上的许多问题的答案并没有完美的“精华”，真正的答案反而如同“捡起来”那样平凡而简单。这时候，困扰我们的，已经不再是一个智力的问题，而是自制力、毅力和决心的问题，而温伯格做的，就是通过一些看似机械却很实用的步骤，培养你的自制力、毅力和决心。

如果你期望成为某个专业领域的领导，或者只是希望自己能有所进步，甚至仅仅是改掉自己的一些坏毛病；我都建议阅读温伯格的《成为技术领导者：掌握全面解决问题的方法》：不要空想，不要慨叹，而是按部就班地捡起自己的身份证。

余晟



# 前 言

Banzan 正在逛市场，无意中听到肉贩与顾客的对话。

“我要最好的肉。”顾客说。

“本店所有的肉都是最好的，”肉贩如是回答，“在这里，你根本找不到次一点的肉。”

听到这里，Banzan 眼前一亮。

——Paul Reps

“Everything Is Best”

from *Zen Flesh, Zen Bones*

这是本关于领悟的书，既有读者的领悟，也有作者的领悟。我自己花在领悟上的时间比逛一圈市场要久得多，仍不敢满足。就拿这本书来说，为写它，我花的时间超过十五年。

那还是 1970 年，我和 Don Gause、Dani Weignberg（我妻子）在瑞士度了个夏天。Don 和我正在写那本关于解决问题的书<sup>①</sup>，而 Dani 在继续她关于瑞士农业社区的人类学研究。过去那些年，我和 Don 一直在研究那些解决问题或成或败的尝试，尤其是与计算机相关的项目。Dani 则在研究新技术引入农业社区的方式。相互沟通之后，我们希望成立一个学习班，为引进新技术系统提供最有力的帮助，可是，最重要的环节在哪里呢？

---

① *Are Your Lights On? or How to Figure Out What the Problem Really Is*，中文版即《你的灯亮着吗？——发现问题的真正所在》，清华大学出版社，2004 年 5 月版。

比较了成功的系统和失败的系统之后，我们很快意识到，大多数成功都依赖少数杰出技术人员的表现。他们有些不断提出革新的技术理念，有些能充分利用其他人的想法。有些人擅长创新，有些擅长谈判，有些人是教师，有些是小组领导（team leader）。与不那么成功的同行相比，他们都具备一种少见的素质，就是能将专业技术知识和领导技巧结合起来。现在我们已经知道了，这类人很擅长创新，此外还掌握了足够的激励和组织技巧，把想法付诸实践。

他们可不是直接从工程学院、科技学院出来的技术专家，也不是管理学院培训出来的那类传统领导。他们属于一种不同的类型——他们是混血儿。这些人的共同特点是关注思维的质量。就像那个肉贩一样，他们希望自己的一切都是最好的。我们称这些人为“技术领导（technical leader）”。

Don、Dani 和我新开办了一个学习班，关注“计算机程序设计中的技术领导力”，第一次活动是应 Dennis Davie 之邀，在澳大利亚举行的。15 名学员中有 14 名评价它是“本人所接受的最有意义的教育”，剩下那位的评价是“本人所接受的最有意义的教育之一。”于是我们知道，我们找到了关键的一环。

之后那些年，Daniel Freedman 和其他一些人加入了我们的团队，学习班在世界各地帮助几百人成为技术领导。当然还包括少数电子和机械工程师，以及其他的培训师。这些人发现，学习班上获得的知识，除去少数关于专业技术的内容外，其他都可以直接适用于自己的工作。于是，我们逐渐删去了技术性的内容，扩大了自己的听众范围。同时，我们自己的视野也在不断拓展。

首先，我们发现，这种形式的技术领导能力也适用于许多与技术无关的问题。我们开始听说，学习班的一些学员把学到的知识应用到了非技术性问题当中。

这些人已经从一般的技术主管转型为解决问题的领导，而这并非运气使然。他们中的许多人并没有察觉到这种转型过程。从表面上看，它们似乎前一天还是技术主管，后一天就成了领导，就好像逛市场的 Banzan 一样。但是如果领导力只能通过突然而神秘的启蒙来获得，那么一个人如何能够成长为技术领导呢？

这些年来，我们获得的最宝贵经验就是，成为领导不靠运气，而靠投入。在学习班上，常会有人好像忽然开了窍，其实我们并没有做太多——肉贩随便说的几句话，就促成了

Banzan 的人生转折。我们的学习班没有教人怎么成为领导，只是加速针对个人的自我成长过程。这本书也是如此：读者权且当它是专为自己开设的领导学习班即可。

在与系统打交道的过程中，我明白了，改变的过程通常是牵一发而动全身的：永远不可能一次只改变一小点。我的一举一动，其实都在重复之前解决某些问题的办法。为了进步，我使用不一样的做法来补充以前的行为。这就好像种子，我已经具备了所有值得培育的行为，剩下的只是择优养育。

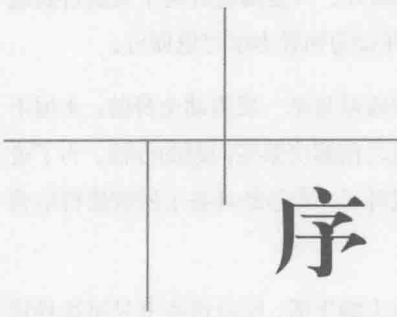
我认为，领导力是一个培育的过程，而不是监管他人的生活，所以这本书只是指导读者驾驭自己的发展。其中的方法就好像我们学习班中的方法一样，不是用来生搬硬套的，而是针对你的具体情况的，它们温和、现实而充满乐趣。

尽管如此，你转变的过程并不会感觉一帆风顺。因为转变是困难的，书中也提供了心理支持的内容。其中那些领导的榜样能帮助你破除先前阻碍自己的迷执。那些转变的榜样能告诉你，旧的观念被淘汰时会发生什么。另外还记录了一些人的叙述，介绍他们成为技术领袖时的感受，这样你就知道自己并不孤单。我知道，你会找到专属于自己的领悟，我希望这本书能贴心陪伴你，逛完整个市场。

G.M.W.

1986年4月

Lincoln, Nebraska



# 序 言

Jerry Weinberg 讲过一个故事：天文学家在花园俱乐部演讲，介绍宇宙起源的“大爆炸”理论。介绍结束，后排一位女士大声说：“年轻人，事情不是这样的。世界其实是驼在一只大龟背上的。”

这样的说法常人无法想象，天文学家却毫不意外，他冷静答道：“那么，大龟趴在哪儿呢？”，女士的回答同样冷静：“显然，在另一只大龟身上。”天文学家觉得自己胜券在握了：“那么，请您告诉我们，这另一只大龟趴在哪儿呢？”。女士平静地笑笑，信心满满，“海龟是一只一只叠起来的。”

Jerry Weinberg 的书很多时候类似他的故事——就好像层叠的海龟。他的书很难一口气读完，因为每一章的内容都像他讲的那些小故事，包含许多层意思。我多次发现自己停下来思考——思考 Jerry 刚刚说过的话，思考我针对 Jerry 的话的思考，思考我针对思考的思考……我想你知道我要说什么。所以我先给读者提个醒：Jerry 的写法往往会引发严谨的思考。

一方面，《成为技术领导者：掌握全面解决问题的方法》是一份非常实用而详尽的指导。另一方面，全书又都在打比方，其中随处可见日常熟悉的事物——讲授管理技巧，靠的是关于弹子球、Tinkertoy 玩具<sup>①</sup>和电热毯的故事。从另一方面来说，这本书讲的是管理技术工程的哲学和心理学。

---

① 一套组合玩具，孩子可以发挥想象力，拼装出各种结构和形状。——译者注

尽管本书令我偏爱有加，但也存在些问题。最重要的问题就是篇幅太长，Jerry 在每一章中开列了太多关于思考和管理的想法和规则，如果你走马观花地看过去——为了写这篇序言，开始我就打算这么干——基本就会一无所获。问题之二是篇幅太短。你以为 Jerry 就要告诉你如何解决世界上的主要问题了，书也就读完了，你发现，其实 Jerry 只是鼓励你自主独立思考。

回头看看，我觉得自己是被标题误导了。我猜你甚至会说，这本书与成为技术领导没什么关系。但是实际上，这本书的核心内容是与 Jerry 所有的书一样的：如何思考，以及如何审视你思考问题的思维。循着层叠的海龟往下，Jerry 指出，在管理和人事协调的实际问题中，大多数显而易见的解决办法其实偏离了问题的核心。所以，他提出简单易行、但又截然不同的办法，来看待我们自认为已经了解的事物。

读者有幸，Jerry Weinberg 把剖析复杂的技术和管理问题——尤其是现代的组织中，两者不寻常的混合体——作为毕生的工作。他的每句话都直指要害。我一而再、再而三地发现，自己一面开怀大笑，一面尴尬不已。

还有一点。任何一篇负责任的序言，都应该为书本推荐适合的读者群。本人当然也会这么做。对于这本书，我诚挚地将它推荐给：(A) 管理者；(B) 被管理者；(C) 认识 A 类或 B 类人群，或是在他们身边的人。如果你有幸落入其中一类，这本书你必须读。

Ken Orr, President

Ken Orr & Associates, Inc

1986 年 6 月

Topeka, Kansas



# 目 录

## 第 1 部分 定义

<b>1</b>	<b>领导力究竟是什么?</b>	<b>2</b>
	领导难当.....	2
	直面领导力.....	3
	领导力：传统但有缺陷的解释.....	4
	截然不同的模型.....	5
	对领导力的全面定义.....	9
	思考题.....	10
<b>2</b>	<b>领导方式模型</b>	<b>11</b>
	激励.....	12
	头脑.....	13
	组织.....	13
	领导力的 MOI 模型.....	14
	技术领导的工作.....	15
	总有更好的办法.....	16
	思考题.....	17