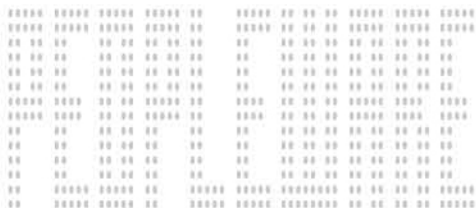


经典  
珍藏

# 人件

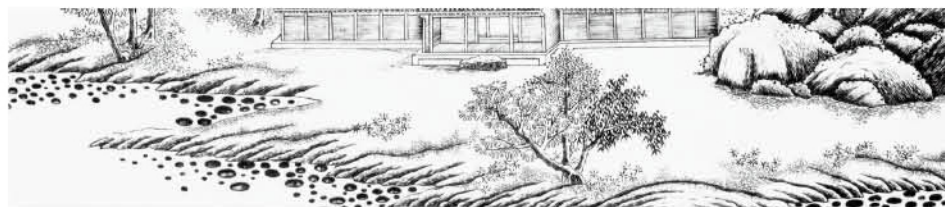
(原书第3版)



PRODUCTIVE PROJECTS AND TEAMS (Third Edition)



[美] Tom DeMarco Timothy Lister 著 肖然 张逸 滕云 译



机械工业出版社  
China Machine Press

名家经典系列

# 人件（原书第3版）

Peopleware: Productive Projects and Teams  
Third Edition

（美）Tom DeMarco, Timothy Lister 著

肖然 张逸 滕云 译



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

人件 (原书第 3 版) / (美) 迪马可 (DeMarco, T.), (美) 利斯特 (Lister, T.) 著; 肖然, 张逸, 滕云译. —北京: 机械工业出版社, 2014.8  
(名家经典系列)

书名原文: Peopleware: Productive Projects and Teams (Third Edition)

ISBN 978-7-111-47436-4

I. 人… II. ①迪… ②利… ③肖… ④张… ⑤滕… III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2014) 第 167686 号

本书版权登记号: 图字: 01-2013-6479

Authorized translation from the English language edition, entitled *Peopleware: Productive Projects and Teams, Third Edition*, 9780321934116 by Tom DeMarco, Timothy Lister, published by Pearson Education, Inc., Copyright © 2013, 1999, 1987 by Tom DeMarco and Timothy Lister.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2014.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

## 人件 (原书第 3 版)

[美] Tom DeMarco 等著

---

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 关 敏

责任校对: 董纪丽

印 刷: 北京画中画印刷有限公司

版 次: 2014 年 8 月第 1 版第 1 次印刷

开 本: 147mm×210mm 1/32

印 张: 9.75

书 号: ISBN 978-7-111-47436-4

定 价: 69.00 元

---

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版 本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

## 译者序

管理与创新这个话题一直困扰着“高”科技行业。前几年的精益创业（Lean Startup）算是论证了创新是必须管理的，如此也在行业里掀起了一股精益创新的潮流。然而，若是对于一家具有相当历史和规模的公司，究竟该怎样管理才能不阻碍创新呢？这仍然是一个难题。时下大家看到、听到的都是一个个昔日巨人被创新颠覆而加速倒下，却鲜有人能够为大家庖丁解牛，分析这样的“大”企业病，很多人仅仅简单地将其归咎为复杂的“企业政治”。

几年前，我在翻看本书第1版时，由于自己阅历有限，读后并没有太多的感悟，甚至觉得这不过是“愤青们”描述的乌托邦。如今拜读本书第3版，却有一种幡然醒悟的感觉。两位大师用严谨的科学实验和辛辣尖锐的解剖一步步为我们揭开了知识经济下管理中的种种误区。反思自己，何尝不是处于“高”科技的假象中，每天习惯性地处理着各种“邪恶”邮件（E-mail），苦恼着为什么只有在



航班上才有独立思考的时间？每个项目总有做不完的需求，成就感却很低，并把按时交付长期寄托于加班这个“奇迹”之上。

掩卷而思，过去工作经历的林林总总，包括初为管理者时的那点小邪恶，好像都被两位大师一一分析成卷，展于世人。我在阅读时，总不免有小小的抵触情绪，但大师十几年的数据收集以及系统的理论分析让人不得不由衷信服。历史学家普遍认为人类是不善于学习的，因为人类总是在重复着同样的经历。两位大师思考的精髓也并非薄薄几百页可以悉数传递的。我们有幸合译此书，在翻译过程中，我们也产生了不同观点，并彼此切磋印证。书中的诸多观点确实让我击节赞叹，但也有部分内容并不认同。作为译者，我们必须忠实于原著；但作为读者，我们不妨批判地进行阅读。正所谓“尽信书则不如无书”，这才是有益的读书之道！

两位大师浸淫 IT 行业二十余年，又精研各种成功、失败案例，对诸多领域皆有涉猎，从而开创了影响软件开发的“人件”理论，可谓筚路蓝缕，立功甚伟。几乎任何一位 IT 从业人员都能从本书中找到自己或者管理者的影子。如果你是“管理邪恶论”的唯技术论者，你会在书中找到印证自己观点的“数据”；如果你是拥有“小小邪恶心态”的管理者，你会在书中得到启示；如果你已经身居要职，困扰于如何从日常琐事中突围，本书会带给你反思；如果你正在引领（或是被迫，不少人这样认为）创新，本书会给予你启发。译者

唯一的分享是：请通读全文，避免断章取义。

人件的想法成于两位大师越洋航班上的一席思考碰撞。在接受翻译任务期间，我因为工作关系总是东奔西跑，翻译间隔时间较长（在此也隔空向两位大师表示歉意）。现在回头来看，发现翻译的大部分工作居然也是成于航班之上！也许如书中所说，万米高空为我们树立起了最好的免扰牌，隔离了邮件和手机的时段让“流”更加顺理成章。这里也要“感谢”商业广告琳琅满目的航空杂志，一来印证了大师们阐述理论的普适性（原来全球化的中国并没有那么不一样啊），二来让翻译（对比阅读航空杂志）成了一件让人身心愉快的事情！

肖然（ThoughtWorks 公司）

成文于北京至成都的万米高空

## 前 言

30 多年前的一次越洋旅途中，在航班上的漫长夜晚，我们萌生了撰写人件相关内容的想法。当时，我们正从洛杉矶赶往悉尼去教授软件工程的系列课程。在飞机上，我们难以入眠，索性谈论起自己经历的以及从客户那里了解到的软件系统的高复杂度。我们中的一人（不记得到底是谁了）从我们的讨论中总结道：“也许……软件系统的主要问题不在于技术，而在于社会性因素。”

我们花了好一会儿才想明白这其中的缘由，这和我们先前对软件的理解完全不同。我们和那些沉迷于高科技的人士一样，坚信技术就是一切，无论出现什么问题，总会有更好的技术为我们找到出路。但是，如果我们所面对的问题天生就属于社会学的范畴，再好的技术可能也提供不了什么帮助。例如，对于一组必须工作在一起的人彼此不信任的情形，就没有什么软件包或万能工具能改变他们，以解决这个问题。

这样的想法一经产生，就驱使我们展开对一些案例的研究，进

而我们俩都认识到，在大多数项目中，社会性的复杂度远比技术上的挑战要难处理得多。而且，不可避免地，我们还要面临一个更加严峻的问题：即便我们意识到社会性因素比技术上的因素重要得多，也从来没有用这样的思维观念管理过团队。是的，我们也会不时地改善团队的协作环境，或者缓解团队的紧张情绪，但这些事情从来没有成为我们工作的核心。

如果我们早些知道人的因素要重于技术因素的话，我们的管理方式会有什么不同呢？于是，我们开始梳理这些想法。正好，我们手头上有空白胶片和油笔，可以将这些令人炫目的想法记录到胶片中，传神而又真实地展现给我们在悉尼的观众。哦，天呐！悉尼可是与美国和欧洲相隔了半个地球那么远，要是我们不回家，谁又会知道我们此时此刻的重大发现呢？

接下来的一周，悉尼的观众立马加入了对人件的讨论行列，当然他们还有那么一点儿懊恼（看来，不只是我们才有唯技术是用论的观点）。最重要的是，人们踊跃发言，分享了不少他们自己的案例，这让我们感激万分，而又备受鼓舞。

较之本书在 1987 年出版的第 1 版，我们通过大量的问卷调查和一线实验证明了第 1 版中关于环境影响的猜测（见本版中的第二部分），同时验证了我们关于团队结构和沟通上的一些更为激进的建议（见本书其余部分内容）。

本书的前两版使我们成了技术项目中人文问题研究的专家，而我们也按着这个思路不断前进。在本版新增的章节中，讨论了一些领导力上的病理症状，在先前版本中这些没有作为病理来归纳；

书中还讲述了会议文化的演进，以及如何管理新旧成员水火不容的混合团队；我们也越来越清楚地认识到，一些日常使用的工具会成为我们前进的阻力而非动力。

对于本书的出版，非常感谢 Dorset House 出版社的 Wendy Eakin 和 Addison-Wesley 出版社的 Peter Gordon 为我们编辑了手稿。此外，还要感谢我们在 The Atlantic Systems Guild 公司的同事——Peter Hruschka、Steve McMenamin 和 Suzanne Robertson，感谢他们 30 年来给我们提供想法，和我们一起进行头脑风暴、开展辩论，为我们提供餐饮美食，以及给予我们的这份浓浓的友谊。

Tom DeMarco 于缅因州卡姆登市

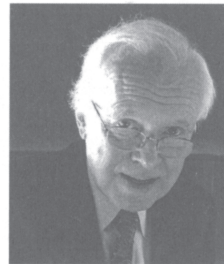
Timothy Lister 于纽约

2013 年 2 月

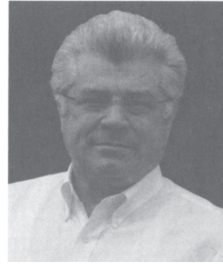
## 作者简介

Tom DeMarco 和 Timothy Lister 是 The Atlantic Systems Guild 公司 ([www.systemsguild.com](http://www.systemsguild.com)) 的首席咨询师。该公司主要提供高复杂度组织结构方面的咨询业务，并且特别关注人在整个组织结构中的重要性。自 1979 年以来，他们俩共同在全球范围内就管理、评估、效率及企业文化发表了一系列演讲，撰写了大量文章，也为很多企业提供了咨询服务。

Tom DeMarco 撰写和与人合作撰写了 9 本书，主题从开发方法介绍到组织结构合理性探讨。另外，他还撰写了 2 本小说和 1 本短篇小说集。他的咨询业务主要集中在专家认证工作方面，另外也常提供有关项目和团队建设方面的咨询。目前，由 Hans-Rudolf Schulz 提供他已经在缅因大学教授伦理学 3 年了。现居住在卡姆登附近。



Timothy Lister 目前主要从事咨询、教学和写作工作。他和 Tom DeMarco 合作撰写了《与熊共舞》(Waltzing With Bears: Managing Risk on Software Projects)，另外他们还和 The Atlantic Guild 公司的另外 4 位同事一起撰写了《项目百态》(Adrenaline Junkies & Template Zombies: Understanding Patterns of Project Behavior) 一书。他是 IEEE、ACM Cutter IT Trends Council 的会员，并且是 Cutter 的研究员。



由 James Robertson 提供

## 译者简介

### 肖然

在从事了多年计算机算法及复杂度研究后，肖然决定尝试变理论为实践，于是开始了软件工程之旅。2006年和开发团队一起第一次运用 Scrum，并开始尝试 XP 的很多技术实践。2008年加入 ThoughtWorks 中国后，他有幸在不同交付项目上系统地学习、运用敏捷的各种实践，并有机会在这个过程中和业内很多思想家探讨了软件开发的各种挑战。在最近两年的工作中，肖然尝试将敏捷、精益思想引入企业管理及组织创新过程中，和组织一起梳理、制定目标，并规划迭代的实施方案。

### 张逸

软件开发生涯经历了从程序员、项目经理、测试经理、开



发部长、技术总监到架构师的一个循环，现在幸福地回到程序员的起点。热爱文学、热爱技术，文理兼修，可惜都不精通。现为 ThoughtWorks 咨询师、程序员。居住在成都，最向往的大约就是在阳光下安静悠闲地品读自己喜爱的图书了。管理上，正大力推进敏捷与精益在软件开发过程中的实践与转型；技术上，热衷于在传统设计领域中引入函数式设计思想。著译包括《软件设计精要与模式》、《WCF 编程》、《Java 设计模式》和《恰如其分的软件架构》。

## 滕云

一个半路出家的程序员；一个军事武器狂热分子；一个并不合格的古典音乐爱好者；一个拙劣的填词人。平时喜欢踢踢足球、吹吹口琴、练练书法等。目前主要从事金融和保险领域的企业级软件开发，主要感兴趣的技术领域包括 Java EE、Linux、领域驱动设计和持续交付等。他的译著有《实现领域驱动设计》。

## 目 录

译者序  
前言  
作者简介  
译者简介

### 第一部分 管理人力资源

**01** 此时此刻，一个项目正在走向失败 // 3

游戏的名称 // 4

高科技的幻觉 // 6

**02** 干酪汉堡，做一个，卖一个 // 7

错误在所难免 // 8

管理：傻瓜定义 // 9

人力商店 // 9

稳定的项目濒临死亡 // 11

我们只是做事，没时间考虑工作自身 // 12

**03** 维也纳在等你 // 14

西班牙理论 // 15

- 来自家里的一句话 // 16
- 不存在加班的谎言 // 16
- 工作狂 // 18
- 工作效率：赢得战斗，输掉战争 // 19
- 反思 // 20

#### 04 质量——如果时间允许 // 22

- 飞离卓越的航班 // 23
- 质量是免费的，但是…… // 26
- 否决的力量 // 27

#### 05 再谈帕金森定律 // 28

- 帕金森定律和牛顿定律 // 28
- 如果经历了我们的见闻，你就不会这样说了 // 29
- 来自新南威尔士大学的数据 // 30
- 帕金森主题的变异 // 33

#### 06 苦杏仁 // 35

- 在睡梦中减肥 // 35
- 七宗罪 // 37
- 这就是管理 // 39

## 第二部分 办公环境

#### 07 家具警察 // 43

- 警察思路 // 44

统一的塑料地下室 // 45

## 08 “朝九晚五在这里啥也完成不了。” // 48

弃权政策 // 49

编码战争游戏：观察生产效率的因素 // 50

个体差异 // 51

生产效率的非相关因素 // 53

你可能不想让你的老板知道 // 54

工作环境的影响 // 55

我们证明了什么 // 56

## 09 在空间上省钱 // 58

席卷大地的瘟疫 // 59

让我们暂停抨击，来谈几点事实 // 61

工作环境质量和产品质量 // 63

诺贝尔奖级别的发现 // 64

躲起来 // 65

## 间奏曲：生产效率度量和不明飞行物 // 67

吉尔布定律 // 68

但是，不知你可否接受 // 69

闭上你的眼睛去度量 // 70

## 10 大脑时间与身体时间 // 72

流 // 72

没有流的无休止状态 // 73

根据流来计算时间 // 74

E 参数 // 76

一座丝巾花园 // 77

对工作的思考 // 78

## 11 电话 // 80

进入另一个世界 // 80

魔界奇谭 // 82

修改过的电话道德 // 84

不兼容的多任务处理 // 86

## 12 门的回归 // 87

表演还未结束，直到胖妈开始演唱 // 87

闪亮的问题 // 89

创意空间 // 90

活力空间 // 92

打破企业常规 // 92

## 13 采取保护步骤 // 94

亚历山大的有机控制理论 // 95

模式 // 98

第一个模式：从工具箱里定制工作空间 // 100

第二个模式：窗户 // 102

第三个模式：室内和室外空间 // 103

第四个模式：公共空间 // 104

模式之模式 // 105

回归现实 // 106

### 第三部分 正确的人

#### 14 霍恩布洛尔因素 // 111

天生与后天练就 // 111

整齐的塑料人 // 112

着装标准 // 114

词汇代号：专业 // 115

企业熵 // 116

#### 15 谈谈领导力 // 117

作为工作压榨机制的领导力 // 117

作为服务的领导力 // 118

领导力和创新 // 119

领导力：言与行 // 121

#### 16 雇一名杂耍演员 // 122

作品集 // 123

技能测试 // 124

组织一场试演 // 126

- 17** 与他人良好合作 // 128
- 首先，机遇 // 128
  - 食物魔法 // 129
  - 是的，但是…… // 130
- 18** 童年的终结 // 131
- 科技——和它的反面 // 131
  - 持续不断的局部注意力 // 132
  - 明确合同 // 133
  - 昔日的杀手级应用 // 134
- 19** 在这儿很开心 // 136
- 离职率：明显的花费 // 136
  - 离职的隐性成本 // 137
  - 人们为何要离开 // 139
  - 一种特殊的病理学：公司搬迁 // 140
  - 永恒之地的观念 // 143
- 20** 人力资本 // 146
- 对人来说呢？ // 147
  - 谁在意这些？ // 148
  - 衡量人力资本投资 // 148
  - 新人上手需要多长时间？ // 151
  - 玩华尔街的游戏 // 152

## 第四部分 高效团队养成

- 21** 整体大于部分之和 // 155
  - 有凝聚力团队的概念 // 155
  - 歇斯底里式的乐观管理 // 156
  - 纳瓦隆大炮 // 158
  - 有凝聚力团队的标志 // 160
  - 团队和团伙 // 161
  
- 22** 黑衣团队 // 163
  - 传奇团队的人员组成 // 163
  - 可怜的地球生物，谁能拯救你们呢？ // 164
  - 小结 // 165
  
- 23** 团队自毁 // 166
  - 防御式管理 // 168
  - 官僚主义 // 170
  - 物理分隔 // 171
  - 时间碎片 // 172
  - 牺牲产品质量 // 173
  - 伪造截止日期 // 173
  - 团伙控制 // 174
  - 重游伤心地 // 175
  
- 24** 再谈团队自毁 // 176
  - 可恶的标语和纪念牌 // 176



加班：一种意外的副作用 // 178

## 25 竞争 // 181

考虑一个类比 // 181

这有关系吗？辅导的重要性 // 182

再谈团队自毁 // 183

混合的隐喻 // 185

## 26 一顿意面晚餐 // 187

团队效应开始起作用 // 187

这里发生了什么？ // 188

## 27 敞开和服 // 190

感觉好，请“病”假 // 190

走出去 // 193

存在规则，但我们要打破规则 // 195

带嘴唇的鸡 // 195

这里谁说了算？ // 197

## 28 团队形成的化学反应 // 198

对质量的执著追求 // 199

结婚时，我告诉她我爱她 // 200

精英团队 // 202

不要拆散洋基队 // 204

团队行为的网络模型 // 204

- 中餐菜谱的选择 // 205
- 做个总结 // 206

## 第五部分 沃 士

- 29** 自我愈复系统 // 209
  - 确定性与非确定性系统 // 209
  - 方法学的隐蔽含义 // 211
  - 疯狂的方法学 // 213
  - 恶意合规问题 // 214
  - 鸡和鸡蛋 // 215
  - 再论高科技假象 // 216
- 30** 与风险共舞 // 219
  - 不要逃避风险 // 219
  - 我们几乎从不管理的一种风险 // 220
  - 为什么不达标的风险总是没有得到管理 // 222
- 31** 会议、独白和交流 // 224
  - 神经硬化 // 224
  - “科技手段增强”的会议 // 225
  - 站立会议 // 226
  - 基本的健康会议 // 226
  - 仪式 // 227
  - 太多参与者 // 228
  - 开放空间社交 // 229
  - 治愈会议上瘾组织的处方 // 231

**32** 终极管理罪恶得主是…… // 232

举例说明 // 232

项目状态会议只关于状态 // 233

早期超编 // 234

再说碎片化 // 236

尊重你自己的投资 // 238

**33** “邪恶”电邮 // 239

忆往昔 // 239

公司内垃圾邮件 // 241

“FYI”到底啥意思？ // 241

是开放型组织，还是公社？ // 242

撤销被动的同意 // 242

建立一个少垃圾邮件、自我协调的组织 // 243

**34** 让改变成为可能 // 245

现在，聆听另一位著名顾问的几句话 // 246

老板，这想法很妙。我马上着手进行 // 248

一个更好的变化模型 // 251

安全第一 // 253

**35** 组织型学习 // 255

经验与学习 // 255

一个重新设计的例子 // 257

组织型学习的关键问题 // 258

- 管理团队 // 259
- 空白地带的危险 // 260

### 36 构建社区 // 261

- 偏离公司政治 // 262
- 为什么需要社区 // 263
- 没有魔法 // 265

## 第六部分 快乐地工作

### 37 混乱与秩序 // 269

- 进步是我们最大的问题 // 269
- 试点项目 // 271
- 战争游戏 // 273
- 头脑风暴 // 275
- 培训、旅行、会议、庆祝和撤退 // 276

### 38 自由电子 // 279

- 小作坊现象 // 279
- 同事、大师、内部创业者 // 280
- 没有前车之鉴 // 282

### 39 霍尔加·丹斯克 // 284

- 然而，为什么是我？ // 285
- 沉睡巨人 // 286
- 醒来吧，霍尔加 // 286

## | 第一部分

# 管理人力资源

**作**为管理者，我们多数人很容易陷入一种典型的失败情境：习惯把人当作固定的模块来管理。当然，这种惯性来源显而易见。回顾我们在走上管理岗位之前所做的准备：我们之所以被认为具备管理者的素质，是因为作为办事员、技术员或开发者的我们所表现出来的良好绩效。这样的绩效得益于我们能够将资源划分为模块，例如软件的过程、电路板或其他工作单元。我们用黑盒的特性来构建这些模块，从而达到屏蔽模块内部特性的目的。设计这些模块，使得它们可以通过标准接口来使用。

由于长年累月对模块化方法的依赖，新晋的管理者很少会怀疑是否能沿用同样的方法对人力资源进行管理。很不幸，这常常并不奏效。

在第一部分，我们一起来探索一种迥然不同的思考人及管理人的办法。这种办法考虑的是怎样去适应人的“非模块化”特征。



## 此时此刻，一个项目正在走向失败

自计算机被广泛使用以来，编写出了数以万计的应收账款程序（Accounts Receivable Program）。当你正在阅读这些文字时，可能又有数十个或者更多的应收账款程序即将完成。然而，此时此刻，一个项目正在走向失败！

想象一下！一个没有真正技术创新的项目正在滑向失败的深渊。应收账款程序不过是一个“重复发明的轮子”，经验老到的开发人员面对这样的项目总能驾轻就熟。即便如此，有时在项目中付出的努力却南辕北辙，最终将项目推向失败。假设其中一个走向崩溃的项目结束，并邀请你前往会诊。（当然，这事儿永远不会发生，我们这个行业自有一条金科玉律来阻止我们分析失败。）

现在假设在所有参与者寻觅到各自的借口之前，你有机会分析到底什么地方出现了差错。自然，你不会将项目遭遇覆顶之灾归因于技术。就当前的技术发展来看，在技术上完全可以实现应收账款系统。一定存在其他因素造成了失败。

在人件（peopleware）项目的第一个十年中，我们对开发项目及其结果进行了调研。我们评估了项目的大小、成本、缺陷、加

速因素以及项目工期的成败。最终，我们统计了 500 多个项目的历史，它们都来自开发一线的项目数据。

统计结果表明有 15% 的项目出现问题：项目取消、终止、延迟或者交付的产品从未被使用。项目越大，出现问题的几率就越高。对于持续时间达到 25 个工作年及以上的项目，足有 25% 的项目最后宣告失败。在早期分析中，我们舍弃了这些失败项目的数据，而对其他项目进行了分析。但自 1979 年以来，我们一直努力联系项目上可以找到的人员，期望发现究竟是哪里出现了问题。我们研究的绝大多数失败项目中，没有一个是因单纯的技术问题导致失败的。

## 游戏的名称

“政治”（politics）是被访问者最常提及的失败原因。但这个词经常被人们习惯性地含混使用。在“政治”这个词语下，包含着诸多不相关联或松散关联的东西，如交流问题、人员安排问题、与上级或客户关系不和、缺乏动力、高离职率等。人们经常用政治来描述所有与人相关的工作，但语言学对这些内容提供了更为准确的描述：它们构成了项目的社会学。真正的政治问题不过是这些病态特征的冰山一角而已。

倘若你认为一个问题属于政治的范畴，你会宿命般地逆来顺受。我们总是能直面技术的挑战，然而坦率地讲，我们又有几人



能自信地面对政治这个圈子呢？认识到问题真正的本质分属社会学的范畴，而与政治无关，能帮助我们面对问题时更游刃有余。项目及团队社会学或许超出了你的专业范畴，却没有超出你的能力之外。

不管你怎么命名这些与人相关的问题，它们都比所有的设计、实现及方法论问题更有可能在下一个项目中给你制造麻烦。事实上，本书的基本论调都是基于这个想法：

我们工作中的问题更多属于社会学范畴，而非技术范畴。

大多数管理者坦承：他们对人的担心更甚于对技术的担心。但他们很少以此种方式去管理。他们的管理方式总是视技术为主要关注点。他们总是越俎代庖，将大量时间耗费在本该由团队解决的复杂而又有趣的难题上，就好像他们是自己完成工作，而非进行管理。他们总是在寻求某种技术银弹（technical whizbang），以期让工作实现自动化（参见第6章）。在他们的职责中，最重要的与人相关的要素却被放到了最低优先级。

滋生这种现象的部分原因来自于管理者的提拔机制。对新晋管理者的训练是如何完成一项工作，而不是如何管理它。很少有公司会考察新晋管理者在工作中是否展示出相应的能力与良好的心态来胜任管理工作。他们缺乏管理经验，也没有具体的实践。那么，新晋管理者又是如何自我说服应该花更多的时间考虑问题的技术因素而非人的因素的呢？

## 高科技的幻觉

问题的症结或许在于高科技的幻觉：广为人知的理论认为凡是接触新技术的人（我们谁不是呢？）就被想当然地看做属于高科技领域。在鸡尾酒会上，当人们畅谈自己就职“计算机行业”、“电讯行业”或者“在线电子交易行业”时，很容易沉溺于这种假象中，认为他们自己就是高科技世界的一部分。在我们看来，他们通常都不是。只有在上面那些领域从事基础研究、获得根本突破的科研人员才是高科技工作者，其他人只是在运用他们的研究成果。我们使用计算机和其他技术来开发我们的产品或者帮助组织我们的事务。由于我们以团队、项目或者其他紧密协作工作小组的形式来完成工作，我们大多数人是在从事人类交流的职业。我们的成功源自于所有参与者良好的人与人之间的互动，我们的失败则归因于这种互动的缺失。

我们习惯性地专注于工作中的技术问题，主因并非它们重要，而是因为它们更简单。安装一块新的硬盘，比寻思为何 Horace 显得忧郁而恐慌，Susan 入职几个月就对公司不满要容易得多。人与人之间的互动非常复杂，没有简单规律可循，但在工作中它的确更为重要。

倘若你发现自己更加关注技术问题而非社会问题，那你就像是一名杂耍演员，在一条昏黑的街道丢失了钥匙，却逡巡至邻近的街道去寻找，并美其名曰：“那里的灯光更明亮。”

## 干酪汉堡，做一个，卖一个

开发的本质完全迥异于生产。然而，开发管理者的思想却通常被生产环境衍生而来的管理哲学所左右。

假设你是一位本地快餐店的老板，那么采用如下任何一条或多条高效生产度量都是合情合理的：

- 压缩出错率，让机器（“人”这台机器）能够尽量平稳地运转。
- 对工作上犯错的员工采取严厉手段。
- 把工人当成是机器上可替换的部件。
- 优化稳定状态。（根本不用考虑运行是怎样开始的，或者需要怎样去终止运行。）
- 标准化流程，让一切有章可循。
- 消灭试验——总部那帮家伙就专门干这事儿。

在快餐行业（或者任何生产环境），这些都是司空见惯的合理手段，但对你来说不是。这种“干酪汉堡，做一个，卖一个”的思维观念在开发领域是致命的。这种做法只能让你的团队士气低落，让他们无法将精力集中到真正的问题上。这种管理风格与开发工作水火不容。

要管理脑力劳动者，你需要与前面走的路背道而驰。我们提出的相反做法会在下面几节详述。

## 错误在所难免

对大多数脑力劳动者来说，工作偶尔出错再自然不过，也很健康，没什么危害。但总有些教条主义者会把工作中的错误和罪恶联系起来。我们需要采取措施去改变这种态度。

面对一群软件开发经理，我们介绍了一种迭代式设计的策略。这个想法是针对那些天生容易出错的设计的，我们应该彻底抛弃而不是去修复。这种设计活动上的死胡同是我们可以预期的，而为此付出的成本却微不足道，不过就是从头再来，轻装前进。令我们感到惊诧的是，许多管理者觉得这是给他们老板出了一道不可能解决的政治难题：“我们怎么能丢掉公司付钱生产的产品呢？”他们好像更相信我们应该补救这个有缺陷的版本，即使从长远来看，我们可能会付出更多。

营造一个不容许任何失误的氛围会让大家持有戒心。他们不愿去尝试那些有可能变坏的事情。当你试图体系化流程时，当你倾向于墨守成规时，你就在强化这种戒心，于是大家就会人为地被禁止做出关键的战略决策，因为他们害怕犯错。在不允许犯错的规定下，或许平均的技术水平会稳步提高，但团队的社会氛围却会遭受可悲的伤害。

相反的做法是鼓励大家犯错。你可以不时问问大家遭遇了哪

些死胡同，并明确地让大家明白：最佳答案并非是“没有”。当有人说了出来，应该祝贺他们——这是他们应得的。

## 管理：傻瓜定义

管理的复杂度使得我们很难简单地定义它，但在伦敦的一次专业学术组织大会上，我们遇到的一位资深管理者让这些细微差异变得荡然无存。他用一句话总结了他对这个主题的观点：“管理就是踢屁股。”这等于说管理者负责全盘思考，而他的手下就照章办事。这种想法可能对于制作干酪汉堡会奏效，但在依靠脑力而非体力的环境中是没有用的。在这样的环境，每个人都要带着脑子工作。踢他们的屁股，可能会让大家行动起来，却不可能让他们去创新、创造以及思考。

即使向人们施压可以增加短期产出，长远来看还是无效的：对于所有工作者来说，若是因为他们感到动力不足而需要老板来“弥补”，没有什么比这更让人沮丧的了。

最可悲的是，这种管理手段几乎永远都会让生产过剩。你根本不需要使用严格的度量来促使大家工作——大部分人是热爱他们的工作的。你有时可能需要采取一些手段让大家少工作一会儿，这样就可以做一些更有意义的工作（更多想法参见第3章）。

## 人力商店

在生产环境中，很容易将人当作是机器的零部件。部件磨损

了，你可以换个新的。新的部件可以和原来的互换。你定的新部件，无论多少，都不过是数字而已。

很多开发经理持相同态度。他们竭力说服自己：没有人是不可替换的。正是因为他们害怕关键人物会离去，于是强迫自己相信没有所谓的关键人物。管理的本质不就是保证工作正常进行，而不用管个体的去留吗？他们的行为看起来就好像有这么一个奇妙的人力商店，只需拿起电话说：“给我派送一个全新的 George Gardenhyer 过来，但让他不要那么高傲。”

我的一位客户在对一名出色的员工进行薪资评估时，惊讶地发现这个员工想要的并不是涨薪。他说他在家时，常常会灵光一现，想起一些绝妙的主意，但糟糕的互联网连接却让他很是心烦。难道公司就不能帮他在家安装一条网线，再给他买台高性能的工作站吗？公司当然可以。在接下来的几年，公司甚至在这位员工家里帮他建立了一个小型办公室。当然，我这位客户的案例有些不同寻常。要是换做一位洞察力偏弱的管理者，他会怎么做呢？太多的管理者认为工作人员展现其个性是一种威胁。

——TRL

有这么一个案例。老板是一名洞察力不够强的管理者，他对员工的个性表现出极端的受威胁感：他有一位富有才华的员工常

年在外到访客户现场，花销自然不少。通过对他报销的分析，显示他在食物上的花销远远超过了其他出差人员——他在食物上的花销超出其他人 50%。在一封愤怒的公开信中，这位老板给这名员工打上了“食物犯罪”的标签。而这位员工的总体花销没有超支，不管他在食物上有多少额外花销，他在其他方面节省了。这名员工并非花的更多，他只是与众不同。

对于盲目尊崇生产世界管理风格的管理者来说，员工的独特个性是一种持续的困扰。人性化的管理者却能认识到正是这种独特性使得项目团队产生了化学反应，是团队充满活力与高效的源泉。这是需要培养的。

## 稳定的项目濒临死亡

稳定的生产思维对项目工作尤为有害。我们很容易忘记项目生命周期的最终目标就是要结束自己。一个项目唯一的稳定期就是将死之时。除非你正在一个被取消或将要取消的项目中，所有的项目管理关注点都应该投入到开发的动态调整上。然而在一个新项目中，我们衡量员工的价值却使用了稳定状态下的特征：他们写了多少代码或者产出了多少文档。我们对于每个员工在整个开发投入中的切合度关注甚少。

几年前，在我教授一堂企业内部设计课程时，一位高级管理者抓住我，要我评估课堂的学员（他项目上的员工）。

他对一位女士尤为好奇，毫不掩饰对她的质疑：“我看不见她给项目带来的贡献，她不是一个好的开发人员或测试人员，或者任何其他专业人员。”在做了一些调查后，我发现了一个引人注目的事实：在她12年的公司生涯中，她所在的项目没有一个不是获得巨大成功的。她的贡献不是很明显，但她所在的项目总是成功了。在课堂观察了她一周，并与她的同事交流过后，我得出一个结论：她就是一个超级催化剂。她的存在使得团队内部更有黏度。她帮助团队成员互相交流和相处，有她的项目会变得更加有趣。当我试图向那位管理者阐述这一理念时，我被震惊了：他居然不知道催化剂这个角色在一个项目中的重要性。

——TDM

催化剂很重要，因为项目总是处于不断变化的状态。一个能够让项目更稳定的人抵得上两个做事的人。

## 我们只是做事，没时间考虑工作自身

如果分配给你一个任务，你会花多少时间来真正实施这项任务？不可能百分之百。实施之前肯定需要做一些头脑风暴，调研新方法，找到规避一些子任务的方法，阅读相关材料，培训，还有试错。

回首我们作为管理者的那些年，我们都没有正确地认识到这



一点。我们都花了太多时间去做事，却没有花足够的时间提出关键问题，“这件事到底该不该做？”稳定阶段的干酪汉堡心态使得我们根本没有去思考这项工作。这种心态会推着我们把百分百的投入放到实施状态。倘若真要为没有思考时间寻找借口，那么这个借口永远都是时间压力——就好像还有什么工作可以在没有时间压力的情况下完成似的。

随着更多利益的介入，思考方法的重要性也显著提高。正所谓“磨刀不误砍柴工”，我们必须学习如何多花时间在思考上，少花时间在实施上。项目需要的投入越夸张，成员就越应该学习如何更好地协作，对这份工作的热爱也会变得更重要。项目越是需要在一个无法完成的固定时间交付，项目团队就越不能缺乏频繁的头脑风暴，或者项目组聚餐之类的活动来帮助团队形成一个统一的整体。

但是，这些都是人文关怀。每个人都知道怎么做，对吗？错。在做事上我们都是如此的一根筋。我们花费不到5%的时间在计划、新方法调研、培训、读书、评估、预算、排期、人员安排这一系列活动上。（5%这个数字来自于对系统开发项目的分析，但这个数字涉及的范围应该更广，可能涵盖了所有拿薪水的工作者。）

关于读书的统计结果让人尤为失望：以软件开发人员的平均水平为例，平均每人没有一本和工作相关的书，甚至没有阅读过一本相关书籍。这种现状让每一个重视质量的人心怀忧虑。对我们这些写书的人来说，简直就是一场悲剧。

## 03

# 维也纳在等你

几年前，我和南加州的一位大项目经理谈到项目中的一场“战争”。一开始，他就谈到了项目成员疯狂工作酿成的恶果。至少两起离婚可以直接归咎于团队成员的加班，还有一位成员的儿子染上了毒瘾，可能就是因为这位父亲太过忙碌而无暇顾及对孩子的教育。最后，测试部门的头儿精神都崩溃了。

在他继续叙述这样的恐怖经历时，我开始意识到此人正在用一种奇怪的方式自夸。你或许会怀疑，倘若再有一个离婚或两个自杀，这个项目就会大获成功，至少在这家伙的眼里事情就是这样。

——TDM

对于大家都说的“聪明地工作”，一个共识就是现实生活中的管理者如何让人们以牺牲自我生活为代价来承受更大的工作强度和更长的工作时间。管理者永远拿着个大喇叭，鼓吹他们的员工加了多少班，并谋算着如何让大家加班更多的手段。

## 西班牙理论

历史学家很早之前就抽象出了不同的价值理论：西班牙理论认为世界上的价值总量是定额的，因而财富积累的道路就是学会从大地或者别人的背上去攫取。另一种英国理论则认为，价值是通过智慧和科技创造出来的。所以，在英国人进行着工业革命的时候，西班牙人推动着他们的轮子来开发新大陆的土地并统治印第安人。他们越洋过海，将大量的黄金搬运回国，而他们的努力换来的却是通货膨胀（黄金太多，而可用商品太少）。

西班牙的价值理论在很多管理者身上都可以看到，特别是在他们讨论生产效率时。生产效率本应该是让单位时间内的工作产生更高的价值，然而它却常被看作是如何在单位付酬的情况下攫取更多价值。二者存在天壤之别。奉行西班牙理论的管理者梦想通过不付薪酬的加班来获得生产效率的提高。无论工作多久，他们都按照每周 40 小时分配，而不是员工现实工作的 80 乃至 90 小时。

这根本就不是生产效率——更像是欺诈——但现在很多美国管理者就在这样做。他们连哄带骗地让员工延长工作时间。他们给员工强化交付日期的重要性（其实可能完全是随意的，世界不会因为一个项目晚了一个月就停止转动）。他们使用各种手段让员工接受那些根本没希望达成的进度安排，让他们因为内疚而不得不牺牲自己的其他时间来工作。他们做任何事情都是希望让员工工作强度更大、工作时间更长。

## 来自家里的一句话

虽然你的员工在办公室里得到的信息是“工作时间长点、强度大点”，但他们在家里得到的信息却迥然不同。家里传递着信息：“生命正在流逝。衣柜里堆着你的脏衣服，你的孩子们得不到拥抱，你的配偶正在寻找别的地方。那种被称为生命的旋转木马只有一圈，中奖的机会只有一次。如果你把你的生活都耗在C++上……”

但你知道何时告知真相，

你或许能够心想事成，又或者你会慢慢变老。

还未跑完半程，你就会被踢出局。

你何时会想到……维也纳在等你？

——Billy Joel, 《The Stranger》

在 Billy Joel 的歌中，那个等你的维也纳就是你人生旅途的最后一站。当你抵达那里，一切结束。要是你觉得你的项目成员从来不考虑这么沉重的问题，那就再想想。你的成员都意识到人生苦短，生活中还有许多事情比这愚笨的工作更重要。

## 不存在加班的谎言

让定薪员工加班是无知的管理人士脑海中的臆想。是啊，星期六工作几个小时可能会对星期一的最后期限有帮助，带来

的后果却是需要花“地下时间”<sup>①</sup>来弥补他们自己的生活。计算投入与产出，每加班一个小时，就需要一个或更多个小时的地下时间。这完全是杀鸡取卵，短期来看或许有利，从长远看其实入不敷出。

疯狂的孩子慢一点，

关掉你的电话玩玩消失。

没什么大不了。失去一两天不会有啥影响。

你何时会想到……维也纳在等你？

类似于不付薪酬的加班时间，对于持西班牙理论的管理者来说，对地下时间也是视而不见的（他们不管人们一星期花费了多少时间，总是以40小时来计算）。在任何人的考勤表里，你看不到它的踪影。这些时间花在煲电话粥，聊天吹牛，又或者休息养神上。没有人能真正工作超过40小时，至少不可能持续，特别是对需要创造力的脑力劳动来说更是如此。

加班就像冲刺：跑马拉松跑到最后100码，体力还有剩余才算有意义。若一开始就冲刺，那你纯粹就是浪费时间。让人们过度冲刺，只能让大家失去对管理者的尊重。最优秀的员工在之前都经历过这种事：当管理者极力鼓吹工作必须在4月完成时，他们目光闪烁，明白此时应该保持沉默。然后，他们只要逮着机会就会补偿性地做与工作无关的事情，到最后，他们

---

① 指虽是工作时间，却做着与工作无关的事情的那些时间。——译者注

真正工作的时间还是每星期 40 个小时。最好的员工都会这样反应，其他不这样的全是工作狂。

## 工作狂

工作狂会做很多不计薪酬的加班。他们的工作时间长得夸张，虽然效率可能在下降。给他们足够的压力，他们会工作到完全丢掉自己的个人生活。但这只能是一段时期，即使是最疯狂的工作狂，信息迟早还是会传递到他们那里：

慢点儿，你做得很好啦，  
这辈子你不可能做你想做的所有事情。  
在这暧昧的夜晚，尽管如此的浪漫。  
但你何时会想到……维也纳在等你？

一旦认同这样的理念，员工就会在完成项目后永远消失。为了不那么重要的价值（工作），而牺牲了生命中更重要的价值（家人、爱情、家庭、青春），这种认识让人备受打击。这会让一个不经意间做了牺牲的人愤怒得想要复仇。他不会去找老板，然后淡定而体贴地劝说事情应该在将来做出改变——另一种情形是他已经心力交瘁，唯有选择辞职。无论哪种方式，他终归是离开了。

工作狂是一种病，但不像酗酒那样只影响不幸的少数人。工作狂就像常患的感冒：每个人都有可能感染上。我们写这些的目的不是要讨论工作狂的产生及治疗，而是指出一个简单的问题，

即作为管理者的你应该如何应对这些工作狂。如果用西班牙理论将他们压榨到底，你最终将失去所有。不管你多么需要大家投入额外时间加班加点，都不能让大家以牺牲个人生活为代价。失去好的员工绝对不划算。这已经超越了工作狂这个狭隘话题，而是复杂而有意义的对工作效率的讨论。

### 工作效率：赢得战斗，输掉战争

下一次，倘若你听到有人提及生产效率，仔细倾听，是否会谈到“员工流失”这个词。他十有八九都不会提到。这些年来，在对生产效率的讨论以及数百篇相关文章中，我们从未听到哪位专家在人员流失这个话题上提出任何观点。但这样的顾此失彼有意义吗？让我们来看看一些组织采取的提高效率的典型手段：

- 对人们加压以获得更长的工作时间。
- 产品开发的流程机械化。
- 牺牲产品的质量（下章我们再具体探讨）。
- 标准化工作程序。

上面任何一项都会让工作减少一分愉快及乐趣。因此，提高生产效率的过程可能会诱发员工去寻找其他更满意的工作。当然，这并不是说我们就一定不能承担因为提高生产效率而造成的人员流失，只是在获得高生产效率时，一定要考虑可能的人员流失，不然所谓的“提高”可能会因重要人员的流失而被抵消。

大多数组织不会对流失人员进行统计。几乎没有人可以告诉你替换一名有经验的员工的成本。无论什么时候讨论生产效率，衡量起来都好像人员流失根本不存在，或者与成本无关一样。Data General 的 Eagle 项目就是这样一个案例。该项目是西班牙理论的胜利：工作狂的项目成员通过不停加班把生产效率推向了一个前所未有的高度。项目最后，几乎所有开发人员都离职了。这样做的成本有多高？没人在计算时考虑这些。

生产效率的定义是收益除以成本。看得见的收益就是通过工作赚取金钱，而成本则是全部的花销，包括替换那些疲惫不堪的工作人员。

## 反思

在我早期的一个咨询项目中，项目运行平稳，项目经理知道她能够按时交付产品。她被叫到管理委员会（management committee）做进度报告。她说她可以保证产品在3月1日的最后期限完成，完全准时地遵循最初的估算。上层管理者仔细地讨论了这意料之外的好消息，第二天又把她叫来。他们解释道：因为她能够在3月1日准时交付，所以最终期限被提前到1月15日。

——TRL



项目实际能够达到的时间表对持西班牙理论的管理者是没有价值的，因为这样的时间表不会把人们放到压力之下。一个更好的根本毫无希望达成的时间表才能让员工们付出更多的劳动。

在职业生涯中，你总会遇到一个或者几个这样奉行西班牙理论的管理者。你可以嘲笑这些人的短视，但你自己也应该反思。我们每个人都在过去某个时刻、不同时期屈从于这种短期策略，给下属施加压力，让他们更努力地工作。为了这样做，我们不得不无视大家效率的下降和人员的流失。要忽视这些负面影响总是很容易，不容易做到的是牢记如下令人不适的事实：

压力不会让人工作得更好——只是工作得更快。

为了工作快点，他们不得不牺牲产品质量以及自身的工作体验。

## 04

# 质量——如果时间允许

20 世纪的心理理论指出，人类的特征被一系列基本天性控制：生存、繁衍、领地等。这些都是直接植入我们脑子里的。你可以不带任何情绪理性地分析这些本能（正是你现在做的），但当你感知到它们的时候，总是会带有浓厚的感情色彩。即使对于这些内在的价值观仅有一点小的挑战也会让人感觉失望。

任何强烈的情绪表达都显示了大脑中的原始价值观受到了威胁。一个新手管理者也许会相信工作可以在不掺杂个人情绪的情况下完成，但只要管理者具备一点点经验，就会知道事与愿违。我们的工作给了我们表达自己情绪的很多机会。

你想想，总有一次某人的情绪完全因为工作相关的事情而被煽动起来。好好思考这件事，然后反问自己（可能是第 n 次了），情绪从何而来？若对这件事的背景一无所知，我打赌，它一定是威胁自信的一大成因。在个人生活中，造成情绪化反应的因素可能很多，然而在工作环境中，主要的导火索就是对自信的威胁。

我们通常倾向于将我们的自信与生产出的产品质量（并非产品数量）紧紧关联。（因为某些原因，生产出大量质量马虎的产品带

来的满足感很小，尽管在某些情况下需要这么做。)采取任何可能牺牲产品质量的行动都可能挑起员工反对你的情绪。

## 飞离卓越的航班

管理人员设定的不可达到的期限威胁着产品的质量，但他们不会这样去思考这一问题——他们自认为给了团队一个有趣的挑战，可以激发他们去追求卓越。

有经验的员工(老油条)却不这么想。他们知道，在枪口下，他们的任何努力都是受到约束的。不能自由调配资源，以满足准时交付的要求。要想增加更多的人，或者减少功能？没门！只有质量是可以降低标准的。当员工处于极度的时间压力之下时，质量就开始被牺牲了。他们会把问题藏到脚垫下面，或者直接扔给产品的最终用户。他们交付的产品是不稳定、不完整的。他们会讨厌自己做的事情，但能有什么选择呢？

一些死板现实的管理者会这样回答上述问题：“我们一些同志会为了‘质量’在一个任务上无休无止，但市场才不管你什么质量呢——昨天就赶着要这个产品了，即使质量粗糙也行。”多数情况下，你对市场的判断或许正确，但通过施压让大家制造出达不到自己质量标准的产品终究是错误的。

我们这些管理者总是认为质量只是产品的另外一个特性，可

以视市场的需求而调整。就像你涂在自制蛋糕上的巧克力酱：想要多点就多涂点，想要少点就少涂点。

然而，产品制造者对待质量却完全不同。因为他们的自信来自于产品质量，所以会有自己的一套质量标准。对他们而言，要让自己满意，最低标准就是要达到过去做到的最好质量。这当然要比市场要求并愿意为之付出的标准更高。

“但市场才不管你什么质量呢。”读到这句话真让人唏嘘落泪，只因事实确实如此。人们可能会大谈对质量的重视，抱怨质量的缺失，但真到了要为质量买单时，他们真正的价值观就露出真面目了。拿一个软件项目为例，你可能对用户解释说：“根据经验证据我们推断，产品目前的故障平均间隔时间（MTBF）为 1.2 小时。要是我们今天就交付，产品会非常不稳定。如果再多花三周时间，预计 MTBF 会增加到大约 2000 小时，这可是一个非常棒的结果。”然后就看到用户开始支支吾吾，含糊其辞。他们会说质量真的太重要了，但三周时间可是真金白银啊。

说到软件，这个行业已经让客户接受了自主开发的应用程序平均 100 行代码就有 1 ~ 3 个缺陷！极具讽刺意味的是，这样灾难性的纪录常被归咎于制造者缺乏对质量的认知。而且，一旦质量降低，那些被责怪做事修修补补永无止境的员工就成了质量问题的罪人。还是让我们把这样的责备转到别处吧！花钱的人说了算，他们已经定下了低质量的基调：把开发流程经常性置于极

度压力下，同时又接受低质量的产品，软件用户群体展现了“真正”的质量标准。

上面所说的一切听起来像是在贬低软件用户或者市场上的通用标准，但事实上我们无须如此。我们需要假定：为我们工作买单的人对质量和成本之间的关系有一个清醒的认知。这里明确的一点是，客户想象的产品质量往往比制造者认为的要低。这是一个天然的冲突。降低产品质量可能会让一些人不再购买，即使靠降低成本来提升单个产品的利润，也无法抵消由于质量下降带来的市场占有率的降低。

让买方而不是制造者来设定质量标准，即我们所谓的飞离卓越的航班。如果我们忽略对制造者的态度及效能带来的影响，那么让市场来产生标准也许是可以接受的。

长远来看，以市场为基础制定的质量标准花销更大。即

质量，远远不只是最终用户的要求，而是达到高产能的一种方法。

如果你对这点有疑问，想象一下下面的试验：到大街上随便找 100 个人，问他们认为谁是以高质量著称的组织、文化或者国家。估计一半的人会告诉你是“日本”。我们再问另外 100 个人：什么组织、文化或者国家以高效能著称？同样，大多数人还是会说“日本”。这个国家在作为质量领袖的同时，也以高产能著称。

等等，高质量怎么可能和高产能并存呢？按常规理论，若要

提高产品质量，就必然需要在生产过程中花费更多。为了找到答案，让我们一起来看看两位最受尊敬的评论家田岛（Tajima）和松原（Matsubara）对日本现象的解释：

价格和质量的对立在日本并不存在；相反，高质量带来成本的降低却是被广泛接受的想法。<sup>⊖</sup>

### 质量是免费的，但是……

菲利普·克劳士贝（Philip Crosby）在他 1979 年出版的《质量免费》一书中阐述了同样的道理。在这本书中，克劳士贝给出了很多案例及理论分析来论证：让制造者来设定他们自己满意的质量标准，会带来生产效率的提高，从而抵消为提高质量而产生的额外成本。

很抱歉，我们认为克劳士贝的著作给行业带来的弊大于利。问题在于多数管理者只听说过书的标题，却没有读过这本书。标题完全以偏概全。随处可见管理者热情地宣扬质量：“只有天空才是质量的上限，我们提升质量是免费的！”这根本不可能形成正面的质量意识，这种态度恰恰是克劳士贝倡导的反面。

关于质量和生产效率之间的关系，需要用一种不同的表达方式传达信息：

---

⊖ D. Tajima and T. Matsubara, “Inside the Japanese Software Industry,” *Computer*, Vol. 17 (March 1984), p. 40.

只有愿为质量倾其所有的人，质量才是免费的。

一个组织，如果为了质量一毛不拔，那么收获的质量也将会一文不值。“质量——如果时间允许”这种策略会导致产品不会有任何质量可言。

很早之前，惠普就被认为是一个通过制造者自定义高质量标准而提升生产效率获益的典型组织了。从一开始，惠普就信奉质量。在这种环境下，需要更多时间或资金来生产高质量产品的说法压根儿就没有出现过。因此，开发人员形成的文化是交付的质量要超越市场的质量要求。他们追求质量的标志成为提高他们工作满意度的动因，从而拥有行业内最低的人员流失率。

## 否决的力量

在一些日本企业，如典型的日立软件和富士的部分团队，项目团队拥有否决发布被认为是未成熟产品的权利。即使客户愿意接受不满足标准的产品，团队仍然可以坚持己见，直到产品质量达标才交付。当然，项目管理者与前面提及的管理者存在同样的压力：他们必须尽快交付。但重视质量的文化由来已久，所以日本的这些管理者知道不能哄骗他们的员工去接受质量上的缺陷。

你能够给予你的团队成员否决交付的权利吗？当然，要迈出第一步需要钢铁一般坚韧的神经。你面临的原则问题在于，帕金森定律会与你做对。这个话题值得我们用一章来讨论。

## 05

# 再谈帕金森定律

英国作家诺斯古德·帕金森（C. Northcote Parkinson）于1954年引入了一个概念，认为工作会自动膨胀，占满一个人可以用的所有时间，这被称为帕金森定律。

倘若你不知道其实很少有管理者接受过管理方面的正规培训，你可能会错以为他们都是科班出身，在学校就学过帕金森定律及其分支理论。即使管理者对管理一无所知，他们也极为认同帕金森定律这一针对员工及其态度进行管理的公理。定律为他们提供了强有力的证据，只有设定不可能完成的交付日期，才能保证工作的完成。

## 帕金森定律和牛顿定律

帕金森定律离公理还差得远呢。牛顿定律才是公理，从这个角度看，帕金森定律显然不是。牛顿是一位科学家，他对重力的研究遵循了一系列严格的科学方法。定律的提出经过了严格的验证和测试，并且通过了接下来几个世纪的后续研究。

帕金森不是一名科学家，他没有收集数据，甚至可能根本就



不知道什么是统计推论规则。帕金森是一位喜剧作家，他的“定律”被接受并非因为它是真理，而是因为它有趣。

当然，帕金森定律还是有真实成分的，要不然就不会那么有趣了。帕金森说，他的定律是通过观察一个虚拟的政府官僚机构得出的，一些人认为原型就是英国邮局。

官僚机构很容易产生这样的问题，因为员工很少从工作中收获满足感。但你很可能不在官僚机构工作，就算你在，你可能也花了不少力气保证你的员工不这么做，否则他们不会有任何产出。结果就是大家都可能从工作中得到莫大满足。因此，一个简单的事实值得澄清：

**帕金森定律基本不可能运用到你的工作中。**

生命苦短，不能在工作中虚度光阴。只要大家热爱工作，就不可能让一项工作变得遥遥无绝期——这会推迟获得他们向往的满足感。在不需要降低标准、牺牲质量的时候，他们和你一样，期望工作能快点完成。

## **如果经历了我们的见问，你就不会这样说了**

每一位管理者在职业生涯中一定会遇到一个员工不想工作，或者完全达不到质量标准，又或者无法完成工作。对此，帕金森定律管用吗？

在一个健康的工作环境里，要是某人表现欠佳，可能是因为能力不足、缺乏自信，或者无法理解项目中别人的想法和项目目标。通过增加进度压力无助于解决上述问题。一旦某位员工表现出工作低效，对工作质量漠不关心，就是发出了明确的信号，说明这个可怜的家伙快被工作困难给压垮了。他需要的不是更多的压力，而是调换工作，或者跳槽到另一家公司。

即使在极端情况下，依靠某人是唯一的出路，管理者也应该将此作为最后的选择。团队发出的声音会更好，我们见证了一些良好团队的管理者和大家站在一起，共同批评那些不与别人协作的员工。

在随后的几章中，我们会就团队以及怎样促进团队良好的化学反应展开讨论。这里，我们不会讨论什么方法有效，而是讨论什么方法无效：把你的团队成员当做帕金森型的员工是不可能奏效的。这只能消磨他们的意志，让他们失去前进的动力。

## 来自新南威尔士大学的数据

当然，帕金森定律的思维观念并不会因为它不应该存在而消失。要帮助管理者认识到帕金森定律并不适用于大多数员工，方法是通过严格收集得到的数据。（且不提帕金森本人并没有提供任何数据来支撑他的定律，仅仅是花了几百页内容来阐述观点。）

两位备受尊敬的新南威尔士大学研究员——迈克·劳伦斯和

罗斯·杰斐逊在上个世纪 80 年代和 90 年代进行了年度调查。他们按照通行的数据收集标准对正在进行的项目进行了度量。每一年，他们会专注于不同的项目工作方向。1985 年的数据调查结果显示：帕金森定律并不具备普适性。虽然结果不能完全推翻定律，但至少提供了足够的理由来质疑。

劳伦斯和杰斐逊设定的目标是比较各种估算方法给产能带来的影响。他们想证明（或推翻）大众认为的开发人员（调查对象是程序员）会更加努力工作去达成他们自己给出的估算。对他们调研的 103 个项目中的每个项目，劳伦斯和杰斐逊计算出了一个产能的加权度量值（weighted metric）。然后他们将项目按照原始的估算方法进行归类。部分结果参见表 5-1。

表 5-1 估算方法的对应产能（部分结果）

工作量估算人	平均产能	项目个数
仅程序员	8.0	19
仅管理人员	6.6	3
程序员和管理人员	7.8	16

至少结果证明了大众的观点：相比管理人员的估算，程序员自己给出估算对应的产能会高一点。若二者一起做估算，对应的产能结果处于二者之间。

在同一年调研的 21 个项目中，估算由第三方完成，通常为系统分析师。在这种情况下，程序员的产能持续高于前面提到的由程序员或管理人员来做估算（见表 5-2）。

表 5-2 估算方法的对应产能（部分结果）

工作量估算人	平均产能	项目个数
仅程序员	8.0	19
仅管理人员	6.6	23
程序员和管理人员	7.8	16
系统分析师	9.5	21

最后一条数据出乎意料，未能证实大众的想法。为什么程序员要更努力工作去达成分析师给出的估算，甚至超过他们自己给出的估算呢？可能我们会觉得这是收集的数据不够准确造成的。但如果你跟我们一样，相信不准确的估算是伤害士气的一个原因，那么其实这些数据并不需要过多解释。系统分析师其实是比程序员或管理人员更好的估算者，他们既了解工作的细节，又不会陷入构建者那种自然的乐观或者老板们通常在政治或经济方面的偏执中。另外，系统分析师通常会有更多的估算经验，因为他们过去做过，而且从中吸取了经验教训，因而能够估算得更加准确。

不准确的估算，或者根本不现实的估算会消磨掉构建者的精力。Capers Jones，以他对开发项目度量的研究成果著名，他这样说道：“当一个项目的时间计划制定得根本毫无道理或不现实，甚至不可能通过加班来达成时，项目团队会变得愤怒和烦躁……他们的士气会跌到谷底。”<sup>①</sup>这种“根本毫无道理或不现实”的计划不管来自于老板还是构建者自己都没啥区别。既然明知不可取胜，

<sup>①</sup> Capers Jones, *Programming Productivity* (New York: McGraw-Hill, 1986), p.213.

人们自然就不会高效地工作了。

在 1985 年劳伦斯—杰斐逊的研究报告中，最后一部分让人尤为吃惊。他们调查的 24 个没有做估算的项目在产能方面远比其他项目强（见表 5-3）。

表 5-3 估算方法的对应产能（全部结果）

工作量估算人	平均产能	项目个数
仅程序员	8.0	19
仅管理人员	6.6	23
程序员和管理人员	7.8	16
系统分析师	9.5	21
（没有评估）	12.0	24

老板不给任何时间压力（“你们做完了告诉我一声就行。”）的项目产能最高。当然，这并不能说明帕金森定律不适用于程序员，但应该足以让你产生怀疑了吧？

是否在一个项目上采取时间进度压力，应该像是否惩罚你的孩子一样来决定：如果惩罚比较少见，而时间点也能够掌握得恰到好处，惩罚就可能起作用。若需要经常施以惩罚，就说明你自身是有问题的。

## 帕金森主题的变异

只要让帕金森定律变异一点，就能在很多组织中造成惊人的事实：

一个组织的工作如果都忙忙碌碌，就会膨胀以至于占满整个工作日。

在一家公司刚成立时，这种现象就开始了，随着时间的推移问题会变得越来越严重。倘若荷兰东印度公司（1651年成立时最大的公司）仍然存在，他们的员工可能会花一周40小时的时间来填各种表格。大家注意，在这种情况下，帕金森式的行为在于公司而非员工。在第二部分，我们会再讨论这个主题。

## 苦 杏 素

苦杏素是从杏树里提炼出来的无色液体。在瑞典，超市里就可以买到这样的东西，价格跟杏仁提取物相当。在烘焙时，你可以像使用其他提取物一样使用苦杏素。然而在墨西哥，你花 50 美金只能买到这么一滴，因为它可以“治愈”致命的癌症。事实上，苦杏素不能治愈任何疾病。所有证据都揭示这其实是一个残忍的骗局，但在走投无路的情况下，无论这种说法多么离谱，病人最终还是会接受苦杏素小贩们的说法。处于绝望中的人们是不会理性去看待证据的。

同理，不少管理人员也“足够绝望”，这种绝望让他们很容易成为某种号称能够提高产能的技术苦杏素的受害者。他们买到的技术其实缺乏任何客观证据的支撑。因为需求如此迫切，他们就会忽略对证据的审视。

### 在睡梦中减肥

一天，我在极度无聊下开始剪报纸上关于能够提升产能 100% 或更多的广告。一会儿我就收集了一大堆。宣传

能提升产能的手段真是花样百出，让人吃惊，有课程、打包项目、方法论、书籍、时间安排表、硬件监测器、计算语言、新闻传单等。那天晚上，我搭乘地铁回城时，在《纽约时报》的背面发现了一个广告，上面说“在睡梦中减肥”，跟其他广告真是异曲同工。

——TRL

我们面临不少需要提高产能的压力。这个问题的解决没有简化之道，因为所有的简单手段在之前就已经被尝试和实施过。一些组织仍然会比其他组织做得好很多。我们相信那些做出卓越成绩的组织并不是采用了什么特别高级的技术。之所以表现良好，就在于他们做到了更高效的人员管理、改进的办公场所以及企业文化，还有就是实施了一些量化的手段。我们将在后面的第二到第四部分进行讨论。

技术手段未必有效，至少短期来看不能立竿见影，这多少让人有些泄气；而我们提倡的对企业文化的改进又很难运用，而且见效缓慢。最理想的是指望剪下杂志最后几页的一张赠券，附上几千元钞票，寄出去，然后就祈祷着从回信中获得一剂灵丹妙药，它能神奇地帮助产能获得提升。当然，这种灵丹妙药帮不了你什么，可这种不是解决方案的简单方案，比起真正复杂的解决方案，往往更加吸引人。



## 七宗罪

不能解决问题的简单技术方案带来的假象就像那些诱惑了奥德修斯 (Odysseus) 的原罪一样，每一项都有着独特的吸引力，却是一无是处的虚假承诺。一旦你迷信这些，就会犹豫着是否该去做那些为建设一个健康企业文化必须付出的辛劳工作。

根据你所从事行业的不同，具体诱惑你的原罪也各有不同。我们从自己熟悉的软件开发领域收集了七种，描述如下，并给出了我们的反馈。

### 软件管理的七个假象

- 有一个你不知道的新窍门可以让产能飙升。

**反馈：**你不可能单纯地傻到对一些基本的东西一无所知。你不停地探寻新方法，尝试可能有用的改进。在你所采用或想要采用的方法中，没有一个能够真正让产能飙升。我们要做的就是让大家能够健康发展：人们愿意全身心参与、去学习、去提高。

在兜售这个点子时，故作高深莫测，采用一种纯粹的心理上的恐吓战术，让你觉得这一神奇创新不容错过。

- 其他管理者正在收获 100%、200% 乃至更多的增长！

**反馈：**忘了吧！这种典型的神奇工具一般只关注整个生命周期的编码和测试阶段。即使我们拿掉编码和测试，也不能期望 100% 的提升，还有分析、讨论、文档、培训、验收、交流和交付

需要完成呢!

- 技术日新月异,你已经过时啦!

**反馈:** 技术确实日新月异,但(回想下高科技幻象)你做的大部分工作并不是真正的高科技。硬件变化确实很大,但软件开发行业却很平稳。我们仍然有很大一部分时间花在低技术含量的需求和文档上。软件业的产能每年也就增长 3% ~ 5%,仅仅比钢铁和汽车行业好一点。

- 改变程序语言会给你带来巨大提升。

**反馈:** 程序语言很重要,因为它们影响着你怎样去思考一个问题。但如前所述,它们仅仅影响了项目的实施部分。这种被夸大的宣传把一些新的程序语言塑造成了苦杏素。确实,可能用 Java 实现一个新功能比 PHP 好,但在 Java 出现前,已经有其他更好的方法来达到你想做的:针对某类功能的专业快速实现工具。除非你在过去几十年都冬眠去了,否则改变一种编程语言是不会给你带来什么根本变化的。可能给你 5% 的提升(不够你一个喷嚏的),不会再多了。

- 因为库存(backlog)的缘故,你需要马上让产能翻倍。

**反馈:** 关于软件开发库存讨论了太多,依旧是一团迷雾。我们都知道一个项目最后的花费总比开始预期的要多很多。可以乐观地假设,一个今年没有构建的系统(因为我们没有人力了)的花费要比实际构建该系统少花费一半或者更多。这是一个典型的在

迷雾般的库存中迷失了的项目，乐观的成本估计只有实际成本的一半，甚至更少。倘若我们知道系统的真正成本，就能看清楚项目的实质：一个经济上的失败之作。项目根本就不应该在库存里，而应该丢到被拒绝的那一堆里。

- 你自动化了其他所有东西；难道不是要你自动化掉你的软件开发人员吗？

**反馈：**这是另一个版本的高科技幻象：相信软件开发人员能够轻松应付可以自动化的工作。然而，他们的主要工作是通过与人沟通把用户对需求的表述变成正式的程序。对于这些工作，不管我们怎么改变，开发生命周期都是必需的，而且不太可能被自动化解决。

- 你的员工在巨大的压力下工作得更好。

**反馈：**他们根本不会——他们更乐于减少压力。

到目前为止，所有这些都是负面的。如果依靠人员会对产能起反作用，采用最新技术来装点系统又不会有什么帮助，那么管理者该怎么办呢？

## 这就是管理

早年，我还是一名开发人员，有幸在莎伦·温伯格（Sharon Weinberg）管理的项目中工作过。她后来成了科德

及日期咨询集团（Codd and Date Consulting Group）的主席。我认为她简直就是启发式管理的活榜样。一个下雪天，我拖着病体搭建着我们那不够完善的系统，准备用户演示。莎伦进来，发现我在控制台前强撑着精神做事。她转身离开，几分钟后，她端着一碗热汤出现了。喝完她给的热汤，我精神一振，然后问她在管理工作如此繁忙的同时，怎么有时间来做这些。她给了我一个招牌式的微笑，然后说：“汤姆，这就是管理。”

——TDM

莎伦熟知所有具有良好本能的管理者深谙的道理：管理者的作用不是让大家去工作，而是创造环境，让大家可以顺利开展工作。