

PH
PTR

人件集

—— 人性化的软件开发

The Peopleware Papers

NOTES ON THE HUMAN SIDE OF SOFTWARE



Larry L. Constantine 著

谢超 刘颖 谢卓凡 李虎 译

 人民邮电出版社
POSTS & TELECOM PRESS

人 件 集

——人性化的软件开发

Larry L. Constantine 著

谢 超 刘 颖 谢卓凡 李 虎 译

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

人件集：人性化的软件开发 / 康斯坦丁 (Constantine,L.L.) 著；谢超等译. —北京：人民邮电出版社，2004.2

ISBN 7-115-11788-8

I. 人... II. ①康...②谢... III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2004) 第 004356 号

版 权 声 明

Larry L. Constantine: The Peopleware Papers

ISBN: 0130601233

Copyright(c)2001 by Larry L. Constantine.

All rights reserved. No part of this publication may be reproduced, Stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior consent of Prentice-Hall, Inc.

Published by arrangement with Prentice Hall PTR. All Rights Reserved.

本书中文简体字版由 **Prentice Hall PTR** 出版公司授权人民邮电出版社出版，未经出版者书面许可，对书的任何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

内容提要

本书是人件领域中的经典著作。本书以专题的形式，探讨了软件开发中的人的因素，包含了广泛的话题。

本书共分9个部分，第一部分——团队开发，6章的篇幅对团队如何工作、如何为开发更好的软件而更好地工作展开了调查。第二部分——男牛仔和女牛仔，4章的篇幅触及了软件开发人员中存在的一些不同观点。第三部分——工作组织，7篇文章探讨了团队组织和组织开发的问题。第四部分——工具、模型和方法，探讨了开发者和他们使用的工具之间的关系，以及他们使用的方法和编程工具之间的关系。第五部分——过程改进，为在过程改进的同时提高软件质量提供了一些建议。第六部分——软件可用性，着眼于软件可用性和用户界面设计问题。第七部分——有用的对象，解释了在用户界面设计和软件可用性方面，使用软件对象和用例方法的相同之处。第八部分——勇敢的新软件，探讨了作为沟通形式的软件所涉及的一些话题。第九部分——文化和质量，论述了软件开发中的组织文化。

本书适合于所有开发并使用软件的设计人员、开发人员和管理人员阅读，本书的许多章节收自于作者在多本知名计算机杂志的人件专栏文章，在带领读者走进人件世界的旅行中，以人性化的角度思考如何迎接软件开发所面临的各种挑战。

2AA37/03

致 谢

本书的主要内容是关于“合作和共同的事业”。本书的出版不是我一个人的功劳，它还包括了许多人自始至终的努力。本书中的绝大部分内容来自我人件专栏中的文章，正是得益于编辑Larry O'Brien和Marie Lenzie的坚持和赏识，我才有机会成为专栏撰稿人。Larry邀请我为《计算机语言》杂志随便写一些我感兴趣的东西。最初我想推辞，因为为专栏写文章就意味着有时间限制，你必须定期写点东西出来，而我不怎么喜欢这种被人“逼债”的感觉。但Larry一直没有放弃，最终，我答应了。Marie将我拉进了《对象》杂志，让鼓吹面向人的我形单影孤地出现在一堆过程和技术专家中。Marie和Larry不仅是优秀的编辑，也是我多年的好友。我曾为Miller Freeman出版社工作过，与Gretchen Bay, Nicole Freeman, Michele Gahee和Nicole Claro共事过。在共同工作时，这些能干的人一直想方设法弄懂我文章中冗长的句子结构，有时还不得不为我那些莫名其妙的邮件而头疼。但正是这些人，向Larry推荐了我。

感谢Paul Petralia, Prentice Hall出版公司的编辑。因为他的热情相助，使这本老书得以新生；感谢Mary Sudul，本书这一版的产品编辑，她对我的书稿进行了非常专业的最终润饰；感谢Harriet Tellem，他对本书第一版的出版也贡献颇多。

除了以上这些一直帮助我的编辑们以外，最后我要感谢Lucy Lockwood，我最好的评论者和顾问。Lucy是我生活、事业上的伙伴，她常常是第一个看到我写出的东西的人，也是第一个能体会到我写作情绪高涨或低落的人。谢谢你，Lucy。感谢所有的人。

前言

软件的另一面

本书谈的是计算机软件的另一面：朝外的一面。技术人员可以接触到（如你、我）这一面，普通的人也可以接触到（如你、我）这一面。本文要探讨的是“人件（peopleware）”的各种不同的表现形式，软件与开发人员之间的界面，软件与使用者之间的界面。

无论是过去刊载这些材料的杂志的编辑，还是Prentice Hall出版公司的编辑们，都同意我扩大探讨的范围。“人件”，包含了如此广泛的话题，使得我几乎能从中找到我所期望的任何话题，并以此来写文章：从组织文化和工程组织、编码混乱和编码纪律、编程工具和编程技术，到用户、可用性和用户界面。人件世界如同一幅庞大的画面，向我展现了一个特别的“中间世界”。在这里，专业技术与社会问题之间形成了一条分界线；在这里，心理学与控制论相遇；在这里，理论与实际相融合。这幅画面反映出长久以来我对人和计算机软件两方面的兴趣，既有个人的，也有专业的。

本书是在（*Constantine on Peopleware*）《康斯坦丁人件集》（Prentice Hall, 1995）一书的基础上进行了修改、扩充和更新。但本书有其独立的主题，主要内容与前者也没有太多相关的地方，所以，不应被看作是前者的增补版。对《康斯坦丁人件集》一书和本书的读者来说，本书收集了更多新的素材。本书既包括我原来在《计算机语言》和《软件开发》杂志“人件”专栏中发表的52篇文章，也包括“完结篇”（参见附录），而且在原有专栏章节的基础上，本书又新增加了25篇第一次发表的短文。作为补充，我在本书又加入了7篇从《对象》杂志中找到的相关题材的文章。这7篇文章对全面理解“实用中心论设计”非常重要。而我在专栏中第一次谈到的方法，经过改进、补充后被写进了获奖书

(Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design 《软件的实用性：实用中心论设计的模型和方法的实践指南》) 中 (参见文献 Addison-Wesley & Lucy Lockwood, 1999)。

写人的最大优越性之一在于，人的变化速度远远低于科学技术的更新。在为本书收集资料的过程中，我重新阅读、编辑了许多文章，我不禁常常发出感叹，在我以软件开发中人的因素方面为题并进行写作的这么多年里，人类社会的变化却是如此之小。方案的实施依然不得不超出预算范围；产品的交付依然没有道理地一再推迟；工作中所需的资料依然难以找到；管理者依然为如何开发、挖掘其下属开发人员的潜在创造力而大伤脑筋。开发人员们则依然为受到设计图表、模型工具和软件开发的种种清规戒律的限制而恼火。反过来，用户们则继续想方设法去搞懂那些计算机一看就明白，但对人来讲却不知所云的软件。

不过，在人类社会没有什么大的变化时，科学技术却发生了急剧的转变，以至于我专栏中最初提及的一些范例和参考现在几乎让人觉得莫名其妙了。举例来讲，我的一篇专栏文章中谈到的“从单色显示器到彩色显示器”，这些，在现在新入行的软件开发人员看来，仿佛是回到了史前时代，不过关于“使用和滥用色彩”的话题还是如同网络一样新鲜。为了保持写作的初衷和原有的风格，我已经对文章中内容进行了相应的更新。

本书共分为九个部分，每一部分包含若干章，每一章中又包含了若干节，原专栏中的文章按照其内容纳入到相应的章节中。关于软件对象和组织文化的可用性的几节也加了进去。读者能够轻易地从书中找到自己熟悉的文章，而且，它们在书中都可独立成章。

本书新增的章有：22~25章，31~32章，40~41章，43~49章以及53~61章，还有附录。

为了完成本书，我到处查寻相关资料，并将它们放入我的文章中。我希望我的书能够保持长久的价值，能不断地鼓励、启发、引导那些希望致力于软件行业的优秀人才们。本书是献给那些开发并使用软件的设计人员、开发人员和管理人员的。这是我写此书的初衷，也是我今天继续进行写作的意图所在。

第一版前言

硬件、软件和人件

单纯使用CASE（计算机辅助软件工程）工具、可视化程序设计方法、快速设计原型或对象技术，并不能开发出一个好的软件。一个好的软件应该出自于“人”，而有趣的是，一个糟糕的软件也同样出自于“人”。1992年，我开始定期为一个专栏写文章，专栏的主题不是关于硬件，也不是关于软件，而是关于人件。这是因为，当时我有一个简单的想法：既然软件是由人创造的，也是由人来使用的，那么只有更好地了解人是如何工作、如何解决工作中的问题、如何协调工作中的关系，才有可能设计、开发出更好的软件。

今天，我们每天都会遇到大量新的词，大多数是旧有词汇的新解，而像“人件”这个词就是罕有的必须重新创造的词。Peter G. Newmann因为他的一份关于“人类的风险与真正的计算机和计算机程序危害”的报告而出名。他应该是第一个正式使用“人件”这个词的人。1976年，他写了一本不太为人所知的书：Peopleware in Systems（《系统中的人件》），在书中的一篇同名文章中，他首次用到了这个词。Meilir Page-Jones在1980年所写的*Practical Guide to Structured Systems Design*（《结构系统设计实践指南》）一书中，再次使用了“人件”一词（正是此书让一般程序设计人员能很好地理解我的作品中关于“结构设计”的内容）。但是直到1987年Tom DeMarco和Tim Lister合著的《人件》一书的出版，才使人件正式成为程序设计领域中的一个专业词汇。

人件，是第三次计算机革命的真正起源地。第一次革命源自“硬件危机”。在一段时期内，人们一直认为自己遇到的计算机问题都源自硬件方面。当时，人们以为，只要有了运行更快、功能更强大的计算机，有更多的内存和更棒的外部设备，就能建立更好的系统，也就能解决所有



的问题。渐渐地，人们有了更好的计算机。年复一年，计算机运行速度越来越快，内存越来越大，外部设备也越来越好用而且便宜，可是计算机问题依然存在。我们仍然在使用运转不稳定的系统，而且，无法及时、有效地在预算范围内完成任务。于是，我们将遇到的问题归咎于软件方面，而第二次计算机革命也随之被称为“软件危机”。人们开始认为，只要有了优秀的编程工具、高级的编程语言、丰富的构件库和辅助程序建立系统，就能解决所有问题，及时、有效地在预算范围里开发出运转良好的软件系统。现在，第三代编程语言变得越来越精密，并出现了第四代编程语言；编译器变得越来越快、越来越聪明；计算机辅助软件工程工具随处可见。结构化革命让我们认识到结构设计和分析，面向对象技术也开始变得成熟、流行。但是我们还是不得不经常改动我们的工作计划，追加预算，计算机问题依然存在。

最后，我们不得不重新认真考虑一下，问题到底出自什么地方？“我们的敌人其实就是我们自己！”是的，人件，就是问题的症结所在。“人”是问题产生的原因，也是解决问题的工具！

人件包含的范围包罗万象。在软件和应用开发过程中，凡是与人有关的任何事物都可以归于人件。我所写的书和专栏中都谈到了人件中所涉及的各式各样的内容：质量和生产率、合作、团队动力、个性和程序设计、方案管理和组织问题、界面设计和人机交流、认知、心理学、思维过程等等。

以上所有话题都是我感兴趣的，也能让我感到兴奋。我当初攻读管理学的部分原因就在于，这门课能让我将计算机、系统原理同心理学联系起来。我的毕业论文就是关于计算机程序设计心理学的。多年来，我已经将心理学家George Miller先生和他神奇的数字（当然是 7 ± 2 了）介绍给了成千上万的学生们和数十位同事了。为了更好地进行软件、应用程序的开发，人们精心设计出结构图表的结构帮助开发人员形成可视化概念，并用于解决相关的问题。接合和连接描述的是人们所看到的计算机程序的效果，它们是结构设计核心中重要的度量尺度。程序设计人员在设计、维护、修改程序时思维过程是复杂还是简单，直接决定到他们设计出的程序是复杂还是简单。

从某种程度上来说，我的工作既不能脱离人，也不能脱离计算机。1976年7月，当美国庆祝独立200周年时，我曾宣布自己告别计算机界，当时，我以为我自此就可以脱开身了。10年间，作为一名受过训练的家庭治疗学家，我的工作对象是夫妻、家庭及有问题的青少年。但是来自业界的压力又将我重新推回了技术前沿。

人件，就是上述提到的技术前沿的十字路口，诸如管理、组织发展、个性、模型、工具、方法、过程、人机交流等方面的问题最终都会体现在人件上。在我写文章时、工作时或教学时，都会不时地提及所有这些方面。为专栏写文章，让我有机会在人件这个广阔的天地中畅游，不时停下来思考一些有趣的想法，直面随时遇到的挑战，在软件和应用开发的大道或乡村小路中信步。

本书记录了我在人件世界中的旅程，从《计算机语言》杂志开始，直到《软件开发》。我做的专栏题目也叫做“人件”。本书中，包含了人件专栏中的所有文章和发表在其他地方的一些相关内容。所有这些短文和文章都已做了编辑，以确保其连续性；其中一些素材，当初为了适应杂志文章长度的要求，做了相应删减，此次出书时经过重新整理又加了上去。当然，这样或那样的改动，都是为了让书中的章节看上去更连贯、更流畅。但是，请记住，本书不是一本百科全书，也不是什么教科书，更谈不上是一份人件世界的路线图，人件世界的疆域实在是太广阔了，本书，充其量只不过是一个旅行者的游记罢了。

我，还将会继续在人件世界的旅行。

作者简介

Larry L. Constantine (拉里·康斯坦丁), 近40年来, 一直是一位软件工程实践和理论领域中的革新者。他是澳大利亚悉尼技术大学计算机科学学院的副教授, 专门讲授软件工程和组织变更管理。他也是 Constantine Lockwood 有限公司研发部的主管, 负责就“以使用为中心”的设计方法进行咨询和顾问。除了著名的《康斯坦丁人件集》一书外, 他还出版了 *Software for Use* (Addison-Wesley & Lockwood), 该书获得了1999年的 Jolt Product Excellent 大奖。

目 录

第一部分 团队开发

引言	2
第1章 决策, 决策	3
第2章 一致意见与折衷	7
第3章 达成一致意见	11
第4章 记录员, 低下还是高贵	16
第5章 办公空间	22
第6章 讨厌的打扰	26

第二部分 男牛仔和女牛仔

引言	32
第7章 牛仔程序员	33
第8章 牛仔归来	38
第9章 多样性的统一	42
第10章 牛仔程序员和软件圣贤	47

第三部分 工作组织

引言	58
第11章 传统战术	61
第12章 混沌方式	66
第13章 开放的结构	72
第14章 花样游泳团体表演	76
第15章 团队策略	80
第16章 因地制宜	84
第17章 反叛同盟	88

第四部分 工具、模型和方法

引言	94
第18章 CASE和认知	97
第19章 关于模型	102
第20章 镜子啊, 镜子	106
第21章 重要的方法	110
第22章 抓住本质	114
第23章 图形时代的来临	119
第24章 软件对象	123
第25章 关于接缝	127

第五部分 过程改进

引言	134
第26章 提高工作的能见度	137
第27章 报酬和重用	142
第28章 超级学习	148
第29章 对瀑布模型进行改进	153
第30章 及时交付	157
第31章 面对压力	161
第32章 体系结构重组	165
第33章 稳步提升的质量	169

第六部分 软件可用性

引言	182
第34章 一致性和常规性	185
第35章 复杂性和蠕变特性	190
第36章 追根溯源	194
第37章 五颜六色的语言	199
第38章 为中级用户着想	204
第39章 英雄无用武之地	208
第40章 编辑界面	212

第41章 服务	216
---------	-----

第七部分 有用的对象

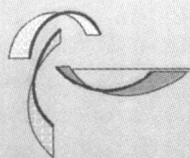
引言	222
第42章 现实对象和软件对象	225
第43章 获取用户信息	231
第44章 抽象对象	238
第45章 新媒体	244
第46章 有用的案例	251
第47章 高效的对象	258
第48章 连贯的对象	264

第八部分 勇敢的新软件

引言	272
第49章 傲慢的程序设计	275
第50章 界面的多样化	280
第51章 向导小精灵	284
第52章 未来的软件	288

第九部分 文化和质量

引言	294
第53章 文化变更	295
第54章 领头羊	299
第55章 最棒的代码是嵌入式的	303
第56章 意大利餐厅的柱子	307
第57章 指导	312
第58章 培训	316
第59章 受奖励的程序员	320
第60章 业界偶像	324
第61章 经理人	329
附录 注册的人件	333



第一部分 团队开发

The peopleware papers

引言

米开朗基罗是一位少有的艺术天才，但西斯廷教堂的建成也并非他一人之功，而是他与其他人合作的结晶。在工作中，米开朗基罗作为管理者，显示出了其艺术天分以外的管理才能。他直呼每一位艺术家或匠人的名字或绰号，鼓励他们发挥主动性，彼此合作，以自己的工作为荣。如果他生活在今天，一定会有人邀请他来参加软件质量管理会议。

软件开发是一种团队开发。无论你把软件开发过程看作是一种艺术创作，还是工程建设，大多数软件都是一组开发人员通过一个共同的系统合作出来的集体创作结晶。一个小组工作的好坏，直接影响到它开发出的软件的质量、决策和主题。如果一个小组合作顺利，对工作中产生的分歧能够统一意见，并能充分有效地利用手中的资源，那么它一定能够开发出好的软件。

团队开发也指一个工作小组的发展状况，指将每个单一个体组合成一个表现出色的小组的过程和实践。许多程序设计人员对团队开发和团队建设心存疑惑。他们对于像啦啦队、团队游戏、团队队歌等这些看上去只是表面上支持团队精神的活动也不愿参与。软件开发团队之所以选择这些“刺儿头”(touchyfeely stuff)，是看中他们的编程技术而不是社交技巧，是编码水平而不是合作能力。

掌握一种编程语言是很重要的，而掌握团队和团队工作过程的语言同样也很重要。因此本书一开始就对团队如何工作，如何为开发更好的软件而更好地工作展开了调查。

第1章

决策，决策

“条条大陆通罗马”，任何事情都可以有多种解决方法！在我整个职业生涯中，我将这个简单的信条作为我的指路明灯，它指引我采用变通的方法来考虑怎样组织软件以及怎样组织人员。但是，我们必须意识到，多种选择也会带来负担，那就是——如何从众多选择中做出正确的决定。开发好的软件就意味着从多种选择中做决策，甚至是从多种选择中挑选出各自的优点对其进行创造性的综合以达到超出各种选择的效果。那些基于“一致意见（consensus）”的方法来做出决策和解决问题的团队可以被认为是组织优良的团队，如果他们知道如何避开那些对于团队来说是常见的陷阱，那么他们可以做出高质量的决定并且可以达到创造性综合的目的。探讨一下这种基于“一致意见”原则进行决策的团队的秘密是非常值得的。

我一直将决策能力作为基本生活能力中的核心能力之一，但若要学会并获得这种能力，除了不断地实践，没有其他更好的途径。也就是说，就像那些成功的家庭和公司一样，都是从众多机会中不断地进行决策实践而产生的。在到达职业生涯中期之前，典型的职业程序员都必须做出几千个决定。很自然的，我们会期望从业人员的这种决策能力越来越好。但是，大部分这些决定都是由程序员个人独立做出的，而在团队中做出决策和解决问题则是由不同的个体共同进行的。

中庸的风险

当我在麻省理工学院的Sloan学校第一次学习管理和团体动力学（group dynamics）时，那些针对基于团队做决策和解决问题的方法所提出的假设性缺陷，耗费了我大部分的学习精力，特别是所谓的风险转移