HZ BOOKS
华章科技

软 件 工 程 技 术 丛 书

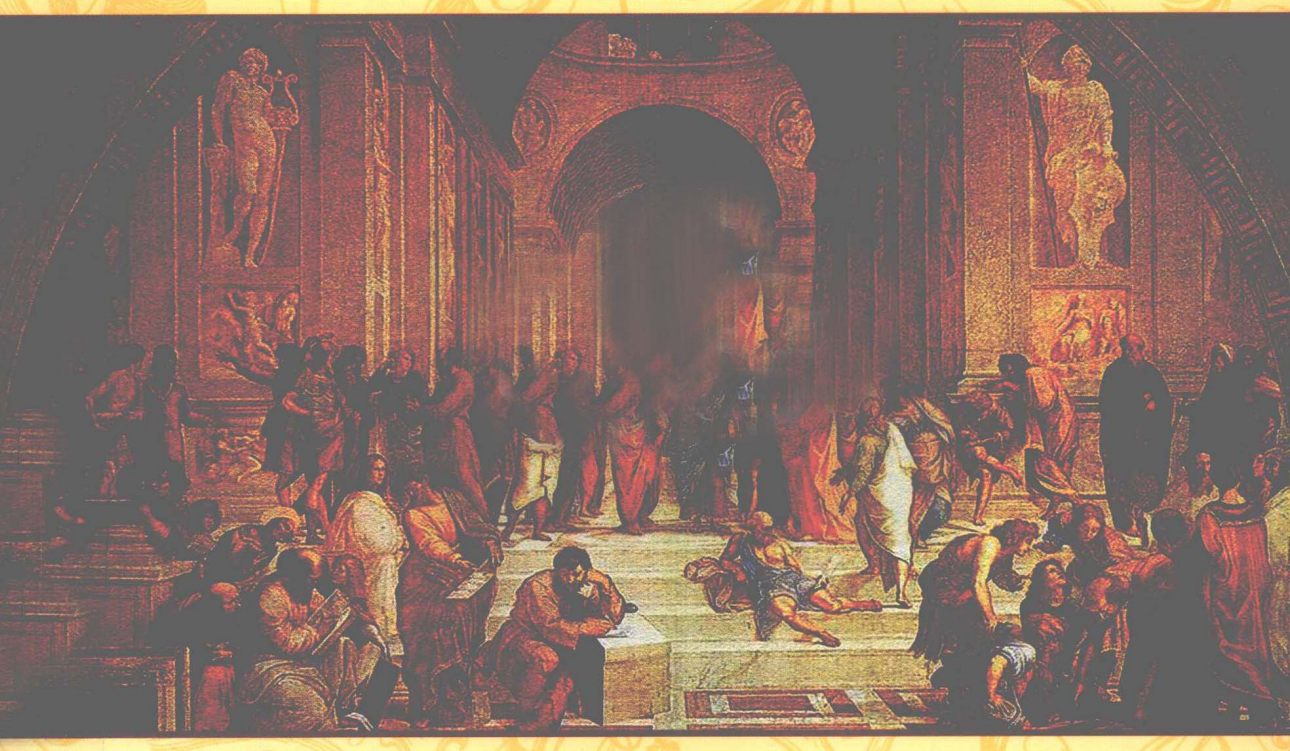
PEARSON

人件集

人性化的软件开发

The Peopleware Papers *Notes on the Human Side of Software*

（澳）Larry L. Constantine 著　谢超 刘颖 谢卓凡 李虎 译



机械工业出版社
China Machine Press

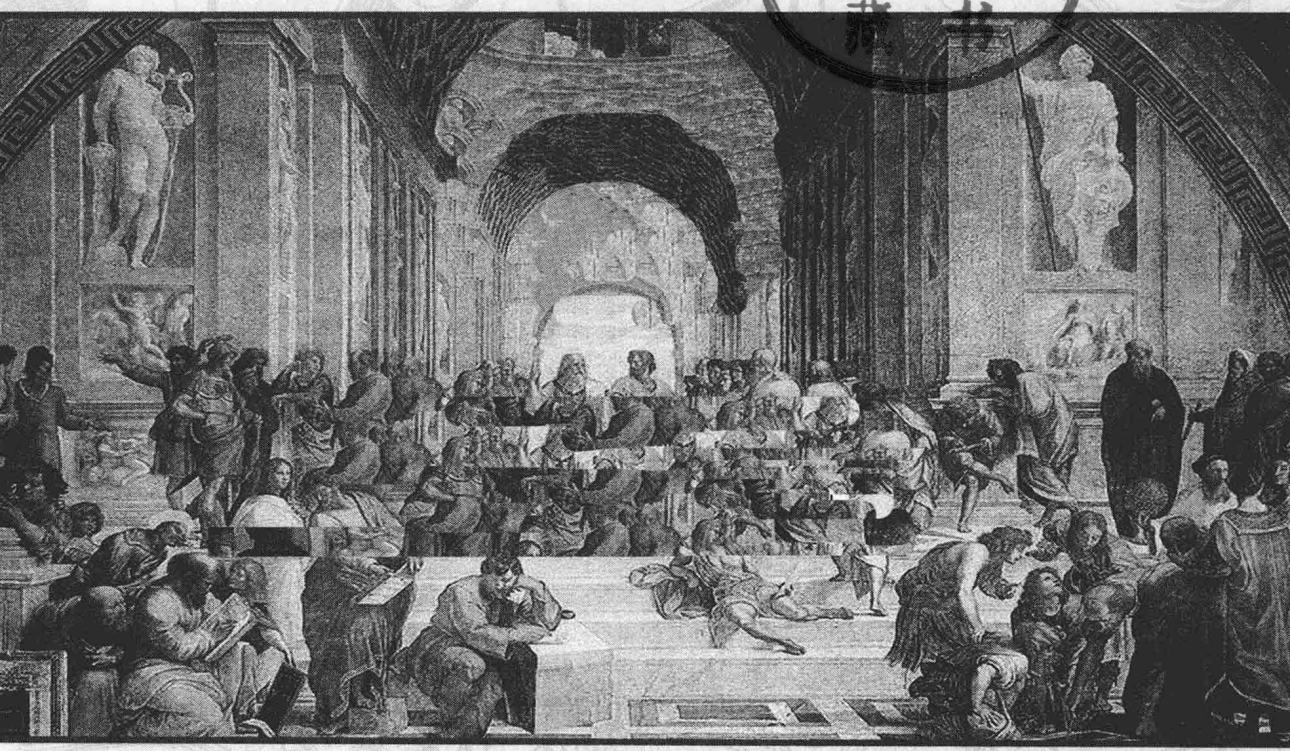
软 件 工 程 技 术 丛 书

人件集

人性化的软件开发

The Peopleware Papers *Notes on the Human Side of Software*

（澳）Larry L. Constantine 著　谢超 刘颖 谢卓凡 李虎 译



机械工业出版社
China Machine Press

本书是人件领域中的经典著作，以专题的形式探讨了软件开发中的人的因素。本书共分九个部分：第一部分介绍团队如何开展工作以及如何为开发更好的软件而更好地工作；第二部分涉及软件开发人员的不同观点；第三部分探讨团队组织和开发的问题；第四部分探讨开发者与其使用的工具之间的关系；第五部分针对提高软件质量提出了建议；第六部分着眼于软件可用性和用户界面设计问题；第七部分解释在用户界面设计和软件可用性方面的相同之处；第八部分探讨软件在沟通中涉及的一些话题；第九部分论述软件开发中的组织文化。

本书的许多内容收自作者在多本知名计算机杂志的人件专栏文章。本书适合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。

本书版权登记号：图字：01-2011-3602

图书在版编目（CIP）数据

人件集：人性化的软件开发/（澳）康斯坦丁（Constantine, L. L.）著；谢超等译. —北京：机械工业出版社，2011.10

（软件工程技术丛书）

书名原文：The Peopleware Papers: Notes on the Human Side of Software

ISBN 978-7-111-36120-6

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（2011）第 206705 号

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：秦　健

北京市荣盛彩色印刷有限公司印刷

2012 年 1 月第 1 版第 1 次印刷

170mm × 242mm · 18.5 印张

标准书号：ISBN 978-7-111-36120-6

定价：49.00 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

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

序　言

硬件、软件和人件

单纯使用 CASE（Computer Aided Software Engineering，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的、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词汇，大多数是已有词汇的新解，而像“人件”这个词却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》，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《Practial Guide to Structured Systems Design》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，有更大的内存和更好的外部设备，就能建立更好的系统，也能解决所有的问题。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高效，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，并及时地在预算范围内交付良好的软件系统。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；可重用构件库得到扩展，编辑软件变得更加上下文敏感，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和流行。但我们还是不得不经常改动我们的工作计划，追加预算，计算机问题依然无处不在。

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类为人件。我所写的书和专栏中都谈到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、项目管理和组织问题、界面设计和人机交互、认知、心理学、思维过程等。

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、系统理论同心理学联系起来。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 和他神奇的数字（当然是 7 ± 2 了）介绍给了成千上万的学生和数十位同事。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以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，思维过程是复杂的还是简单的，直接决定了他们设计出的程序是复杂的还是简单的。

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，我自以为可以就此脱身了。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到技术前沿。

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、人机交互等方面的问题最终都会体现在人件上。在我写文章、工作或教学时，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，还可以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上信步。

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，到《软件开发》杂志结束。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内容相关的文章，所有这些短文和文章都已经过编辑处理，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，此次在本书出版过程中经过重新整理又加了上去。当然，这样或那样的改动，都是为了让书中的内容看上去更连贯、更流畅。但是，请记住，本书不是一部百科大全，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的世界中旅行。

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。本书要探讨的是“人件”（peopleware）的各个方面——软件与开发人员之间的界面、软件与用户之间的界面。

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，都同意我扩大探讨的范围。“人件”包含了如此广泛的话题，以至于我几乎能从中找到我所期望的任何话题，并以此来写文章：从组织文化和项目组织、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。人件世界如同一幅巨大的画面，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的界线是模糊的；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，也有专家的。

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，所以不应看做是前者的增补版。相对于《康斯坦丁人件集》一书，本书收集了更多的新素材。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），而且在原有专栏内容的基础上，本书又新增加了 25 篇第一次发表的短文。作为补充，我在本书中又加入了 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“面向使用的软件设计”非常重要。而该方法（首次提出是在我的专栏中）经过改进、补充后写进了获奖图书《Software for Use：A Practical Guide to the Models and Methods of Usage-Centered Design》（面向使用的软件设计[⊖]）中（Addison-Wesley，与 Lucy Lockwood 合著）。

⊖ 本书中文翻译版已由机械工业出版社出版，ISBN：978-7-111-34575-6。

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速度。在为本书收集资料的过程中，我重新阅读、编辑了许多文章，不禁常常发出感叹，在我以软件开发中人的因素方面为题进行写作的这么多年里，人类社会的变化如此之小。项目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；工作中所需的资源依然难以找到；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。开发人员则依然为受到设计图表、建模工具和软件开发的种种“清规戒律”的限制而恼火。反过来，用户则继续想方设法去弄明白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，以致我的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的过渡”，这些在现在新入行的软件开发人员看来，仿佛回到了史前时代，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，我已经对文章的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相应的章节中。关于软件对象的可用性和组织文化的几节也加了进来。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。

本书新增的内容有：第 22 ~ 25 章，第 31 ~ 32 章，第 40 ~ 41 章，第 43 ~ 49 章以及第 53 ~ 61 章，还有附录。

为了完成本书，我到处查找相关资料，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。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。这是我写本书的初衷，也是我今天继续进行写作的意图所在。

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“面向使用为中心”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，与Lockwood合著），该书获得了1999年的Jolt生产力大奖。

目　录

第六部分 软件可用性

第七部分 有用的对象

第八部分 勇敢的新软件

第九部分 文化和质量

第一部分
Part 1

团队开发

引言

虽然米开朗基罗是一位少有的艺术天才，但是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，充分显示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，并以自己的工作为荣。如果他生活在今天，一定会有人邀请他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做是一种艺术创作，还是工程建设，但是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。许多程序员对团队开发和团队建设心存疑惑。他们对于像拉拉队、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”，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调查。

第 1 章
Chapter 1

决策，决策

“条条大路通罗马”，任何事情都可以有多种解决方法！在我的整个职业生涯中，我将这个简单信条作为指路明灯，它指引我采用变通的方法来考虑怎样组织软件和人员。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，并对其进行创造性的综合以达到超出各种选择的优势效果。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看做是组织优良的团队，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，但若要学习并获得这种能力，除了不断实践外，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，它们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解决无数个问题，因而要做出几千个决定。很自然地，我们期望从业人员的这种决策能力越来越强。但是，大部分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，分析那些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（risky shift）和团体反倾向（counter-tendency of groups）这两种缺陷，以致我对于团队决策的理解达到中庸的地步。在那个保守的年代，就算是那些有民主主义思想的管理者，最

为担心的问题也都是风险转移，而不是蔓延的中庸主义。直到 20 世纪 70 年代发生剧变，团队思想才成为了时代思潮。

研究表明，集体的决策比从集体中的个体独立做选择更具有风险倾向。如果将这种决策模式应用于软件编程，我们可能会看到这样的结果：团队可能使用更奇特的数据结构、更古怪的算法或者更晦涩的语言来编程，这样做必然会给这个软件带来风险。然而，团体动力学的研究还表明了另外一种倾向：在做决策和解决问题时，团队有一种均衡的效果，这会将个人的贡献和能力降到最低。从上面两种研究可以得出结论，似乎单独做出决策更具有效力。其实这两种倾向都是可以尽量避免的。

上述两种倾向在团队决策中存在的可能性在很大程度上都依赖于团队的组织和领导方式。以前，我曾在国外一家新公司中与顾问和管理者一同工作，在那里中庸思想占据主导地位。对于许多在旧体制下训练出来的管理人员来说，团队合作意味着大家均处于一个无能的工作水平上。在这种管理体制下，团队是一种逃避责任的方法，有时他们积极地合谋来限制成绩。在这样一个团队中，挺身而出或者提出一个新颖的观点或者有争议的想法，或者任何出头的行为，不仅仅要冒招致同事怨恨的风险，还可能被人牢记在心，并期望你今后都能达到如此的成绩。在典型的“大锅饭”体制下，如果考虑“饭碗”的稳定性以及“多劳不多得”的分配制度，谁愿意给自己惹麻烦呢？

一个团队所处的社会和组织的大气候是真正发挥团队潜力的决定性因素。为了达到最好的效果，社团文化和团队领导应该积极鼓励和支持创新与协作。从某种意义上说，上述团队做得很好，达到了老板和企业政策制定者“真正”期望，即掩盖事情黑暗面而不是获得好的结果。该团队的管理者告诉我，他们都学会了“绝不要成为链条上的最后一环”。

轻度领导

在“一致意见”方式的设计和决策制定时，团队领导所扮演的角色是至关重要的，不仅要从宏观上建立起一种协作的气氛，而且还要在微观上起到领导的作用。要想使“一致意见”方式的设计和决策制定达到最佳效果，必须让最终的解决办法体现所有成员的智慧，包括经验、创造性以及各种思想，它应该不仅仅是个人贡献的一个均衡产品，而是一个真正综合了各成员最优贡献的产品。不管团队的领导者多么具有能力，当他开始压制成员自身的行为时，团队的工作质量就下降了。

领导团队需要方法，有些方法看上去很狡猾，甚至可以说是“阴险的”。作