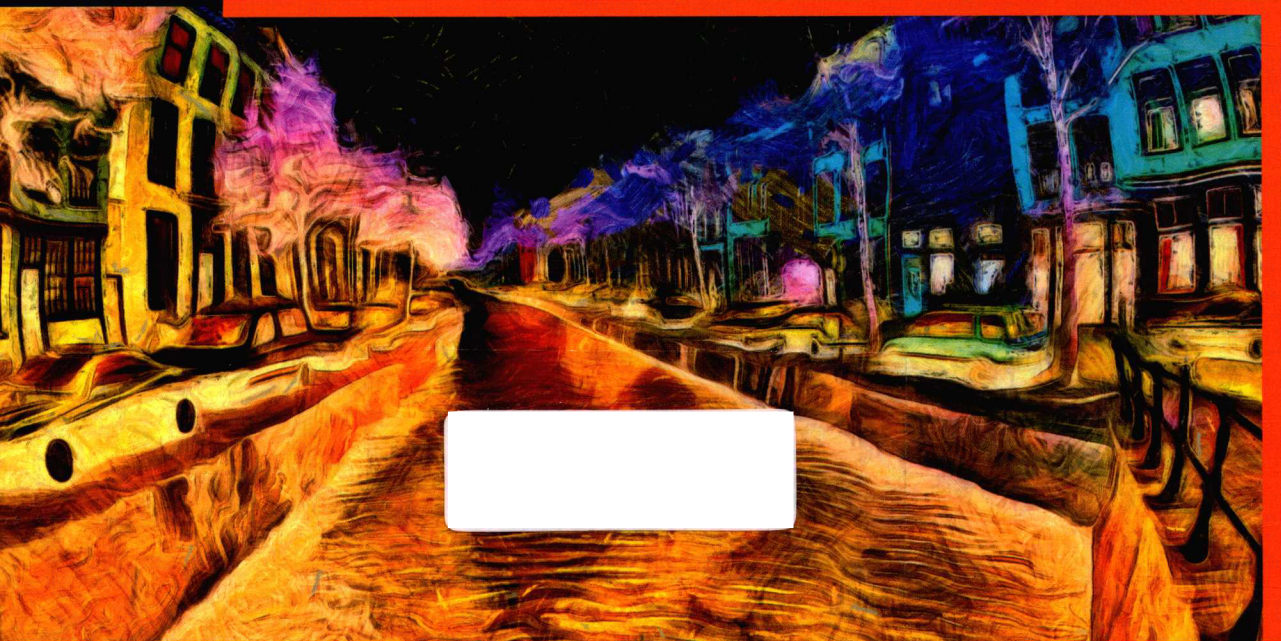


软件开发生产率改进

软件管理的有效领导力与量化方法

Improving Software Development Productivity

Effective Leadership and Quantitative Methods in
Software Management



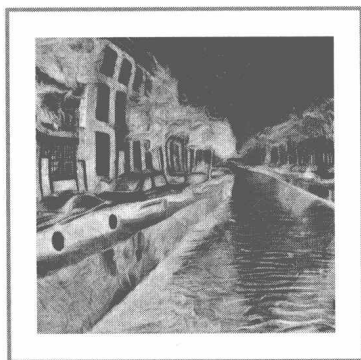
[美] 兰达尔 W. 延森 (Randall W. Jensen) 著

吴舜贤 译



机械工业出版社
China Machine Press

华章程序员书库



Improving Software Development Productivity
Effective Leadership and Quantitative Methods in Software Management

软件开发生产率改进

软件管理的有效领导力与量化方法

[美] 兰达尔 W. 延森 (Randall W. Jensen) 著

吴舜贤 译



机械工业出版社
China Machine Press

图书在版编目(CIP)数据

软件开发生产率改进: 软件管理的有效领导力与量化方法 / (美) 延森 (Jensen, R. W.) 著; 吴舜贤译. —北京: 机械工业出版社, 2016.1

(华章程序员书库)

书名原文: Improving Software Development Productivity: Effective Leadership and Quantitative Methods in Software Management

ISBN 978-7-111-52476-2

I. 软… II. ①延… ②吴… III. 软件开发—管理 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2016) 第 000087 号

本书版权登记号: 图字: 01-2014-8007

Authorized translation from the English language edition, entitled Improving Software Development Productivity: Effective Leadership and Quantitative Methods in Software Management, 978-0-13-356267-5 by Randall W. Jensen, published by Pearson Education, Inc., Copyright © 2015.

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 © 2016.

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。

软件开发生产率改进 软件管理的有效领导力与量化方法

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: 陈佳媛

责任校对: 董纪丽

印 刷: 三河市宏图印务有限公司

版 次: 2016 年 1 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 16.75

书 号: ISBN 978-7-111-52476-2

定 价: 79.00 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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@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

对本书的赞誉 *Praise*

“大多数进步的组织都会采用先进的技术或者严密的流程，力求提高他们的整体效益。而往往被忽略的是那些开明的领导者，他们能够激励并构建一个能够充分利用组织所引入的新技术和新流程的环境。某些流程和技术能够让优秀的团队更卓越，某些流程和技术可以防止平庸者误入歧途。正是开明的领导者告诉我们哪些方法是必需的，然后构建起一个能最大限度提高团队绩效的环境。本书也讨论了这些能力，并把它们有效地包含在评估组织生产率潜力的量化过程中。”

——格里高利 H. 米克尔逊，美国雷神公司集成防御系统部

The Translator's Words 译者序

软件行业里，生产率度量历来都是非常困难的事情。首先，没有合适的度量指标来衡量软件的规模和研发的工作量。软件行业并不存在诸如“每小时生产一辆汽车”这样明确、令人信服而又普遍适用的软件生产率度量指标。人们最常用的源代码行指标，由于种种弊端，饱受诟病。其他指标，比如用例、用户故事、对象点、特性点等，由于种种原因，适用领域非常有限。虽然功能点方法可横向对比，但难以使用，仍未广泛普及。敏捷方法普遍使用最流行的用户故事，但由于用户故事基点定义的随意性使得敏捷项目的基准度量变得几乎不可能。别说不同项目/团队的横向对比，就连同一个团队不同时期的纵向对比都非常不可靠。

其次，即便存在历史数据，研发团队的人员变动之大，也使软件团队的能力始终处于变动之中。人是敏捷开发中的核心要素，但也是最难以度量的因素。无论是老员工离职还是新员工加入，都会让团队的效率在短时间内出现较大波动。人的性格、过往经验以及对开发语言、环境、工具、流程、编程规范等的使用和理解千差万别。对于同一个开发任务，不同的开发人员、开发组织使用不同的实践方法，所得出的软件规模、工作量、成本和进度估计也相差巨大。开发组织的能力同样难以直观地进行量化度量和比较。

再者，软件研发是研发人员的脑力劳动，是智力活动的外在表现。软件行业的历史也就几十年，而软件行业的很多实践和方法均来自建筑和工程行业。与建筑和工程行业相比，软件研发更注重人的智力活动。正如德鲁克所说的，IT行业是知识密集型行业，IT从业者是知识工作者。以往工程实践的管理方法并不完全适用于知识工作者。敏捷实践的兴起、自组织、跨职能团队的流行、Y理论管理理念的大行其道，都说明管理理念的变革势在必行。软件规模、生产率和开发工作量度量，本质上都是对人的智力活动和产物的度量。

正如作者在书中所分析的，几十年来，软件行业出现了各种各样的银弹，每种银弹都声称成倍提高生产率。而事实却并非如此。兰达尔的研究与另一位行业领袖卡帕斯·琼斯（Capers Jones）对软件开发生产率的研究结果基本一致，近几十年来，软件行业并不存在能成

倍提高软件生产率的银弹。

“如果你不能度量它，你就没法儿管理它。”尽管软件生产率度量和改进困难重重，但仍有很多人迎难而上，并做出了卓越的贡献和非凡的成就。兰达尔 W. 延森 (Randall W. Jensen) 就是这些“明知山有虎，偏向虎山行”的勇者之一。在本书中，基于过去几十年的研究和实践，兰达尔详细阐述了一种有效度量软件开发者能力的概念性模型及其商业实现模型，仔细分析了影响软件开发生产率的各种要素，研究了度量软件规模的有效指标，从管理和沟通的角度分析了人员和环境在软件项目中的重要性。作者给出了一套度量软件开发组织能力、软件项目成本和进度估算、有效规模估算方面的可靠量化方法，并结合具体示例，给出了可行的改进措施和建议。

无论是软件开发人员、项目经理，还是软件组织的 CTO 等管理人员，阅读本书都会给你带来很多具有启发性的想法。译者阅读本书，对比自己曾经历过的团队、项目和工作环境，常常会有一种恍然大悟的感觉。在阅读、翻译及与作者讨论的过程中，深切地感受到国外研究者在软件开发基础领域的扎实研究功底。他们从软件项目数据的收集、整理、分析和总结中去芜存菁、抽丝剥茧地寻找软件开发的本质，透过重重迷雾，认真、详细地提取能够改善项目生产率的各种因素，逐步建立数学模型并为将来的估算和规划提供量化指导。这种严谨的科学精神也值得中国的 IT 从业者和研究者学习。

兰达尔是国际知名的软件工程度量大师，是软件生产率度量领域的先行者之一。从 20 世纪 50 年代开始尝试提高生产率的各种措施。通过在休斯飞机公司的工作，他开创性地建立了概念性有效性公式，从管理、技术和沟通三个方面度量个体员工和组织的有效性。而后又提出了 Seer、SEER-SEM 等模型，尝试量化软件项目涉及的方方面面，定量度量软件组织的能力、对生产率的影响、项目成本及进度估算等。本书正是兰达尔工作、研究和咨询的结晶。

兰达尔是一位非常有耐心的长者。在翻译的过程中，译者每次遇到不明白或不太理解的地方，都会给 Randy (兰达尔的昵称) 发邮件询问。Randy 每次都很认真、细致地予以解释和澄清。有时候我们还会来回讨论具体的内容。本着真实传递作者核心思想的原则，译者尽最大努力忠实于原书，尽全力准确翻译，对某些词句反复推敲和研读。另外，作者在书中所说的“能力计算器”电子表格在 InformIT 的网站上需要用英文书籍的 ISDN 注册后才能下载。译者向 Randy 索要了该工具，并获得授权可以分享给需要的读者。如果需要该工具，也可给我发邮件。同时也可登录华章网站 (www.hzbook.com) 下载。

限于译者的知识水平、经验资历、英文理解和中文表达能力所限，再加翻译时间仓促，存在翻译错误在所难免。如果读者在阅读的过程中对本书的内容有任何疑问，或者发现任何地方的翻译并不准确，或者对有关软件项目生产率度量和改进的话题感兴趣，希望就这些话

题进行讨论，均可发送电子邮件到 wushxian@hotmail.com。译者会仔细阅读每一封邮件并逐一回复。译者将整理翻译方面的问题，并在重印版本中予以更正。敬请广大读者提供宝贵的反馈意见，以帮助译者进一步提高今后译著的质量。

感谢郭茵熙、赵晓辉、侯景慧、孙挺挺、孔喜梅认真阅读了翻译稿并给出了大量中肯的改进建议。感谢陈佳媛编辑、吴怡编辑的帮助和她们为本书所做的辛勤付出。

IBM 中国开发中心全球化项目经理 吴舜贤

2015 年 6 月于北京

wushxian@hotmail.com

作译者介绍 *About the Author*

兰达尔 W. 延森 (Randall W. Jensen)

1955 年，当还是电气工程专业的一年级学生时，兰达尔 W. 延森就首次开展了生产率改进实验。在休斯飞机公司空间与通信事业部工作时，他获得了博士学位，并想方设法提高生产率。同时，他被要求开发一个计算机模型来估算大型软件开发项目的成本和进度。完成后的计算机模型提供了一个工具，软件开发管理人员可以用这个工具来度量组织的能力，定量预测与人员、管理方式和开发环境有关的管理决策对生产率的影响。他和查克·托尼斯 (Chuck Tonies) 在 1979 年 Prentice Hall 出版社出版的《软件工程》中发表了生产率的关键属性——沟通、管理和技术。在这 3 个属性和本书中所阐述内容的基础上，兰达尔在生产率领域的多年研究最终让他在 1990 年建立了一个改进的基于计算机的估算模型。

吴舜贤

IBM 中国开发中心全球化项目经理，CSM、CSPO、PMP。2005 年吉林大学计算机硕士毕业后在微软从事软件测试工作，参与并带领团队测试了多款重量级软件，包括 SQL Server 2008。2009 年加入 CA，参与和带领团队测试了多款企业级 IT 管理软件。2012 年加入 IBM 软件全球化团队，从事 IBM 软件全球化、本地化等方面的工作，带领团队与全球 30 多个国家和地区的测试人员合作完成多个软件国际化项目。目前专注于软件全球化、敏捷项目管理、软件质量与测试。软件工程译作 2 本。

Preface 序 言

有些书只需浅尝，有些书可以狼吞，有些书则要细嚼慢咽，慢慢消化。

——弗朗西斯·培根

《论读书》

生产率 (productivity) 是对某种产出 (output) 与所需投入 (input) 的数量或质量的度量。在某种意义上，生产率就是效率和 (或) 质量的度量。幸运的是，在过去的 50 多年里，软件行业持续地以每个人月所交付的源代码语句数量的方式度量软件开发生产率。这里所说的语句是实际交付的代码语句数，而不是开发期间所编写的语句数。源代码语句是以软件开发者所使用的编程语言来描述的。在开发过程中，很多语句编写出来之后随即就被丢弃了。实际交付的语句可能包括多种语言，比如统一建模语言 (UML) 和 C++。以编程语言编写的语句，即使是 1 个 UML 语句也可能相当于 40 个 C++ 语句。投入的工作就是产生单个代码语句所需要的工作量。

我对生产率改进的兴趣最早始于 1955 年，那时我还是犹他州立大学 (Utah State University) 电气工程专业的一名学生。当时要完成繁重的全日制四年本科课程，还从事着一份兼职工作，对校园政治也充满了兴趣，同时又在一个大学生联谊会担任职务。我自己的时间管理不足以应付这些繁忙的工作，所以我不得不想办法降低课后作业和实验室报告对我的影响 (提升生产率)。显然，我发现了一个既简单而又符合逻辑的办法来提高自己的效率。数年之后，我应用同样的办法，一边攻读电气工程博士学位，一边全职教书，一边经营着一份咨询业务，还写了两本书。我可不认为自己有什么杰出的才智，也不是一名优秀出众的学生。我只是在解决问题的时候，受到了“三个臭皮匠，赛过诸葛亮”这一理念的影响。它使我终生对组织管理充满了兴趣。

20 世纪 70 年代中期，我开始认真研究管理技术及其在软件开发成本和进度方面的影响。

早期的研究成果让我相信，软件开发中的人力方面对软件生产率和质量有着重大影响。我在软件开发团队上的第一次认真实验始于 1975 年，在生产率上取得了 175% 的提高并减少了近 3 个数量级的错误。几个同事告诉我，1975 年的实验是首次有记载的结对编程实践。

查克·托尼斯（Chuck Tonies）和我在 1979 年出版的教科书 *Software Engineering*[⊖] 中写道，软件工程师对于组织的价值 V 依赖于有效性公式的 3 个属性：沟通技能 C 、管理理念认知 M 和技术能力 T ，即：

$$V=C[M(T)]$$

毫无疑问，有效性公式已经成为当今那些广为流行的敏捷开发方法的基础。它也是传统软件开发方法中质量和生产力改进的基础。人、激励（motivation）和沟通是所有成功项目的关键因素。

本书的第一个目的就是要探究那些推动高产环境的主要因素和理念，提供评估组织开发环境有效性的手段，揭示项目经理们在其软件开发项目初期和执行期间所做出的决策对生产率的影响。

在过去的几十年里，很多技术变化都已成为业界的标准规范，而生产率的影响却远远超过了这些技术变化的影响，并且包括了更多富有戏曲性的影响，它们都是有效性公式中管理和沟通属性的一部分。

我将探讨有效性数值评级的组成部分，以及能够在组织里提高这些评级并改善组织的生产率、有效性和估算能力的手段、方法。同等重要的是，我会让你对开发流程及管理方式与环境的相互作用有一个更好的理解，以使你能极大地提高开发生产率。除了使用环境参数来计算开发成本和进度，这些度量指标还可以用来评估任何管理决策对环境的影响。

本书可以帮助回答关于项目环境的某些问题，比如“办公室隔间的使用耗费多少项目成本？”“选择从 Ada 变更为 C++，改变项目编程语言的成本是多少？”或者“依靠将我们的能力成熟度评级（CMMI）从 3 级提升为 5 级，生产率能提升多少？”

本书的第二个目的是，建立一个估算方法，用以在各种各样的项目和环境条件下，在经过训练的分析师手中，能够产生软件开发进度与所需资源的现实可行的估计。

很明显，本书最有可能的读者就是软件开发人员。这里的软件开发人员包括所有的管理人员和对生产率提升感兴趣的专业人士。这里所描述的理念，也可等效地应用于敏捷和传统软件开发。沟通、团队协作和环境的运用成为结对编程（Pair Programming）的一部分，而结对编程正是敏捷开发的一个实例。

⊖ Jensen, R.W., and C. Tonies, *Software Engineering*, Englewood Cliffs, NJ: Prentice Hall, 1979.

由于软件行业产生了超过 50 年的丰富文档资料和历史数据，软件开发成为解释生产率相关概念的最理想工具。

当我尽力编制将会从本书所展示的材料中获益的读者列表时，我回忆起了第一个受益人，一个苦苦挣扎尽力完成电气工程专业大学学业，同时又要兼职工作的学生。本书中所讨论的所有关于沟通、团队协作和环境的概念都是能够使人梦想成真的原因之一。基于这一点，我扩展了本书中各种想法的受益者名单，不仅包括软件经理、工程师和程序员，还包括任何需要人员介入并有效、高效使用这些人员的行业（信息技术、制造业、通信、教育等）的人员。20 世纪初期著名的霍桑实验将这些想法应用于装备制造业并取得成功。因此，这些想法真正具有普适性。

正如汤姆·狄马克（Tom DeMarco）和蒂莫西·李斯特（Timothy Lister）在他们所著的《人件》^①中所讲述的那样，软件开发流程是以人为中心的，而不是自 20 世纪 50 年代就开始流行的传统的以技术为中心。尽管传统开发流程依然是当今一些大型组织中使用的主要开发流程，但本书提供了一个路线图，定量地支持软件组织从传统向现代软件开发方式与环境转变。如果要想实现生产率和质量的大幅提高，全面质量管理（Total Quality Management, TQM）理念，包括软件开发团队和 Y 理论（Theory Y）管理的运用，都成为应该考虑的显而易见的管理技术。然而，这种转变并不全是牛奶和蜂蜜。任何过程转变都会付出短期代价，无论是最新、最先进的计算机辅助软件工程（Computer-Aided Software Engineering, CASE）工具、新的开发语言、CMMI，还是开发团队方法。

我在软件成本与进度估算领域的工作始于 1978 年支持一个大型空基软件系统的投标。我的任务是为休斯飞机公司的空间与通信事业部（我的雇主）开发一套仿真模型，仿真结果将用作他们软件开发估算的基准。我建立了一个数学模型，通过使用包括开发人员能力和项目实施约束在内的环境参数，生成准确逼真的估算结果。从远自 20 世纪 60 年代^②开始以来的多个数据来源累积的完整的软件开发项目数据帮我创建了这个数学模型和本书中使用的组织能力计算器。令人惊讶，甚至是令人害怕的是，这个在 1980 年制定出来的数学模型仍然能够应用于当今的软件开发方法且产生准确的工作量与进度估算结果。

本书的大部分篇幅都会探究这样一个简单概念及其对软件开发生产力的重要性。简单地讲，开发环境的生产率及其导致的软件开发成本与进度，取决于 3 个重要属性的驱动：沟通、

① DeMarco, T., and T. Lister, *Peopleware* (New York, NY: Dorset House, 1987)。(中文版《人件》(原书第 3 版)已由机械工业出版社于 2014 年 9 月出版发行。——译者注)

② Wolverton, R.W., "The Cost of Developing Large-Scale Software." *IEEE Transactions on Computers*, June 1974.

管理和技术。尽管本书聚焦于软件，但书中的原则也可不加修改地应用于其他领域。

本书主要由两部分组成：有效领导力和定量方法。

有效领导力

本书前 7 章重点关注有效领导力和组织能力度量。

- **第 1 章，软件开发问题。**该章讨论“软件危机”和自 20 世纪 60 年代以来持续用来解决软件危机问题的技术方法。尽管人们在软件工具、语言 and 开发方法上已经取得了巨大进步，但从 1970 年到现在，软件生产率的提高却非常缓慢。
- **第 2 章，有效性公式。**该章讨论有效性公式的 3 个属性（沟通、管理和技术）在生产率改进中的重要性。同时也会讨论有效沟通的机理及一些阻碍软件开发有效提升的重要文化问题。
- **第 3 章，软件管理的重要性。**该章探讨有效开发中的两个人事管理原则——霍桑效应（Hawthorne Effect）和 X 理论 /Y 理论管理原则。这些原则也是现代人事管理的基础原则。该章还会举例探讨敏捷软件开发与这些原则的相互关系。
- **第 4 章，从历史中我们学到了什么。**该章讲述我们从有关软件开发生产率、技术对生产率的贡献、CMMI 对生产率的影响及乐观的开发预算和进度所产生后果等历史经验、教训中学到了什么。
- **第 5 章，软件开发团队。**在该章中，我们会探讨一下软件开发团队。好的团队、差的团队、丑陋的团队以及他们对软件生产率的影响都会有所讨论。
- **第 6 章，组织能力度量。**该章介绍软件开发生产率度量的一个流程，该流程可应用到你的软件开发组织以揭示组织的内在能力、恒定不变的基础技术以及组织在行业中的相对位置。该章还包括一个用于支持上述评估的工具。
- **第 7 章，完美软件公司案例研究。**该章将介绍能力评估的一个研究案例，总结本书前 7 章的所有内容。

定量软件开发管理

本书剩余部分将会讨论定量管理在成本和进度约束条件下的软件产品交付中的应用。

- **第 8 章，产品复杂度。**该章分析软件系统复杂度在软件开发成本与进度约束方面的影响。

- 第 9 章，人员配置问题。该章介绍软件开发中最佳开发人员配置的一个评估方法。最佳人员配置是产品复杂度的函数，与软件开发方法无关。
- 第 10 章，Seer 软件模型介绍。该章介绍用于软件开发定量管理的 Jensen 软件模型，该模型也是 SEER-SEM 和 Sage 估算工具的基础模型。
- 第 11 章，开发环境。该章将讨论软件开发环境对产品成本和开发进度的影响。环境评估包含与经验、不稳定性和其他管理约束有关的考虑因素。
- 第 12 章，产品特性。该章详细论述产品特性和需求等约束条件在生产率和软件开发环境方面的影响。
- 第 13 章，开发进度与成本估算。该章通过介绍完美软件公司研究案例讲述软件开发成本与进度的估算流程。这些估算受组织能力及环境等施加约束条件的限制。
- 第 14 章，有效规模估算。有效规模并不是新写或修改的软件源代码行数的简单估计。该章探讨了预测开发成本与进度估算所需要的有效规模。
- 第 15 章，功能点规模估算。该章介绍传统软件产品开发中的功能点规模估算方法。作为基于对象软件开发方法规模估算的一种替代方法，该章还讨论对象点的使用。
- 第 16 章，维护估算。维护估算可不是简单的软件增强估计。作为软件产品支持的重要补充，软件维护方面知识留存的影响在该章也会有所讨论。
- 第 17 章，总结。该章回顾本书前 16 章中所介绍的信息，并将这些概念应用到非软件开发环境中。
- 附录 A，软件估算模型。该附录讨论定量软件评估模型的演化历史以及每一种主要估算方法的估算能力。
- 附录 B，参考读物。该附录包含一个广泛的补充读物列表，涵盖有效领导力（沟通、人力管理方法和问题以及技术）、定量管理和估算等主题。
- 附录 C，名词术语。该附录列举本书中所使用的名词术语的一般定义。
- 获取能力计算器。全书用来支持组织能力计算及软件工作量和进度估算的能力计算器（Capability Calculator）电子表格可从华章网站（www.hzbook.com）上免费下载。

致谢

我要感谢为本书的写作提供帮助的人。首先感谢休斯飞机公司空间与通信事业部（SCG）的查克·托尼斯（Chuck Tonies）和肯·哈伯德（Ken Hubbard）的贡献，他们支持我开发出 Jensen 模型，鼓励我展开试验，这些试验已成为本书所讨论的管理理念的实践基础。感谢我

的项目经理弗兰克·沃尔夫 (Frank Wolfe) 对我在软件估算研究方面的积极支持, 以及空间与通信事业部的执行经理丹·福斯特 (Don Forster) 授权我负责估算技术, 鼓励我将 Jensen 模型商业化并开展软件估算咨询工作。在我研究 Jensen 模型并开展估算咨询的冒险之旅中, 常年和我一起工作的另一位关键人物是苏珊娜·卢卡斯 (Suzanne Lucas)。

4 位来自希尔空军基地 (Hill AFB) 美国空军软件技术支持中心 (USAF STSC) 的工程师: 莱斯·杜派克斯 (Les Dupaix)、马克·伍尔西 (Mark Woolsey)、汤姆·罗杰斯 (Thom Rogers) 和布伦特·巴克斯特 (Brent Baxter), 他们是我在软件技术支持中心工作软件估算团队的成员和贡献者, 也是所有估算、培训和技术推广方面有价值的协作者。

最后, 感谢我的妻子玛吉 (Marge), 感谢她多年来对我本书写作生涯的所有支持。为完成本书, 我们只花了 25 年的时间。

兰达尔 W. 延森 (Randall W. Jensen)

Contents 目 录

对本书的赞誉

译者序

作译者介绍

序言

第1章 软件开发问题	1
1.1 软件危机	2
1.2 生产率的人为影响	7
1.3 敏捷对开发生产率的贡献	10
1.4 魔术弹	11
1.5 软件开发的约束因素	12
第2章 有效性公式	13
2.1 有效性公式	14
2.2 沟通的机理	15
2.2.1 信息对流	17
2.2.2 信息辐射	18
2.2.3 四条沟通戒律	18
2.3 文化问题	24
2.3.1 组织文化	24
2.3.2 舒适区	25
2.3.3 确认偏误	26

2.3.4 K-12 现象	28
第3章 软件管理的重要性	31
3.1 历史回顾	32
3.2 霍桑效应	34
3.3 X 理论 /Y 理论	36
3.4 敏捷软件开发	37
3.5 更多有效的生产率改进办法	38
3.5.1 技术	39
3.5.2 开发环境的信息流	40
第4章 从历史中我们学到了什么	42
4.1 潘多拉与魔瓶	42
4.2 技术带来的生产率增益	43
4.3 软件能力成熟度模型集成	47
4.4 最大与最小进度	49
4.5 更多思考	50
第5章 软件开发团队	52
5.1 软件开发团队	52
5.2 团队示例	53
5.2.1 Ad Hoc 团队	54
5.2.2 首席程序员团队	54
5.2.3 结对编程团队	55
5.2.4 跨功能团队	60
5.2.5 老虎团队	62
5.3 团队的积极影响	63
5.4 团队规模	65
5.5 更多思考	67
第6章 组织能力度量	69
6.1 开发者能力概念的演变	70
6.2 有效性公式回归	73

6.2.1	分析师和程序员的能力	74
6.2.2	应用领域经验	77
6.2.3	现代实践方法的使用	79
6.2.4	现代工具的使用	83
6.2.5	终端响应时间	85
6.2.6	硬拷贝周转时间	85
6.3	组织能力评级	86
6.4	基本技术常量	89
第7章	完美软件公司案例研究	94
7.1	问题	94
7.2	完美软件公司	95
7.3	软件组件	96
7.3.1	任务规划	96
7.3.2	命令生成	97
7.3.3	通信模块	97
7.4	完美软件能力评级	98
7.4.1	分析师和程序员的能力	98
7.4.2	应用领域经验	99
7.4.3	现代实践方法	100
7.4.4	现代工具	100
7.4.5	终端响应时间	101
7.4.6	硬拷贝周转时间	101
7.4.7	完美软件能力评级	101
第8章	产品复杂度	103
8.1	量化复杂度值	103
8.2	完美软件复杂度案例	107
8.2.1	任务规划系统	107
8.2.2	命令生成系统	108
8.2.3	通信系统	108