Sun microsystems

Software by Numbers

Low-Risk, High-Return Development

“对基于价值的、与金融运作相关的软件工程来说，本书可谓一个巨大的贡献。”

—Barry Boehm 博士

USC Center for Software Engineering 的主任

COCOMO 和螺旋模型的创始人

价值驱动的软件开发

(美) Mark Denne　Jane Cleland-Huang　著　　肖国尊　译

清华大学出版社

TP311.52
90

TP311.52
90

价值驱动的软件开发

(美) Mark Denne
Jane Cleland-Huang 著

肖国尊 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2004 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: Software by Numbers:Low-Risk, High-Return Development , by Mark Denne, Jane Cleland-Huang, Copyright © 2004

EISBN: 0-13-140728-7

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as **Pearson Education.**

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由培生教育出版集团授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号　图字：01-2003-8073

版权所有，翻印必究。举报电话：010-62782989　13901104297　13801310933

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

图书在版编目(CIP)数据

价值驱动的软件开发/(美)丹尼(Denne,M.)，(美)克里兰德(Cleland,J.)著；肖国尊译.
—北京：清华大学出版社，2005.1

书名原文：Software by Numbers:Low-Risk, High-Return Development

ISBN 7-302-09526-4

Ⅰ. 价… Ⅱ. ①丹… ②克… ③肖… Ⅲ. 软件开发 Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字(2004)第 094435 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969

组稿编辑：曹　康
文稿编辑：于　平
封面设计：康　博
版式设计：康　博
印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市新茂装订有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×230　**印张**：13　**字数**：218 千字
版　　次：2005 年 1 月第 1 版　2005 年 1 月第1 次印刷
书　　号：ISBN 7-302-09526-4/TP・6653
印　　数：1～4000
定　　价：30.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770175-3103 或(010)62795704

前　言

“您绝对应该写这本书!”

这就是我的经理，Stu Stern，Sun 公司专业服务中心副总经理对这本书的热情回应。这本书吸收了应用开发方法中的最新观点，并将这些方法主要应用于获取经济利益而不是技术利益。

尽管我们在创造开发方法这一过程中作出了努力并取得了成功，但只有少数创业者认识到了利用这些技术使经济利益最大化的潜力。毕竟经济利益在软件开发中通常是成功的永久度量，至少在商界中是如此。

本书采用了近几年来在赢得几个富有竞争性的系统集成和应用开发项目的合同方面的经验。虽然赢得这样的合同无疑是利用创新的方法在竞争中取胜，但是同样事关利润的获取。价格必然存在于用户的预算之中，那么投资者要想得到高额利润，就必须尽量降低开发成本，并且必须在应对风险时调整利润值。这些观点并无新意，它们对于任何竞争采购都是正确的。

有关软件开发的不同之处是我们恰好学会了理解价值创造。最普遍的看法是软件开发要承担风险和损失。尽管如此，即使最顽固的反对风险的开发机构都会意识到软件开发会带来固有的价值。如果不能创造价值，就没有人会投资软件开发。遗憾的是，开发机构的所有创造热情和商业热情通常都集中于降低成本和风险。无论是在投标阶段还是在实际运作阶段都是如此。开发人员运用最新的软件方法，创立最新的项目管理策略，并且不断地发展风险缓解技术，主要只是为了实现一个目标：即控制成本。

相反，大部分的运作过程对客户来说是不可见的，在这些范围内，客户很少参与作出重要项目决议的商讨。

在 20 世纪 90 年代，我为东南亚某国政府做大规模的竞争性采购工作。这个项目的本质是，由于评估参数被紧紧地束缚住了，因此技术创新过程中的区别就很有限，我们得用另外一种不同的方法赢得竞争。最后我想到，如果我们对将价

值返还给客户的时间进行优化，而不是仅仅关注控制风险和成本，也许能够提出一种与众不同的价值方案。于是，我们根据价值单元，对客户的要求进行重新分析和分类，最后发现我们确实可以调整开发次序，这样一来，与优化总成本相比，我们就可以更快地交付实际价值。同时，我们可以将成本分为更多可控部分来分期偿还，每一部分都与它的利润值有关。

客户的介入是很有影响的。此方案能有效地降低借款底线和利息支出，便于产品更早地进入市场，并且依据经济前景创造一个更好的模式。

商业银行需要重新计算项目经费数目，开发人员需要理解为什么我们明显要对用户的要求重新排序，当然，客户自身也要看到、理解并会计算这种方法带来的好处。在这段时间内发生了极不寻常的讨论。开发人员就投资利润的问题参与到与银行家的讨论中；项目经理将数据分析表与金融家和投资者进行交易；分析家根据“价值的转化”而不是根据功能效率来衡量此体系结构；在这个过程的结尾，不需要分别向客户做技术方面和经济方面的两种陈述，只需给出一种涵盖这两方面的陈述，包括这一组讨论所表现的所有方面。我们赢得了时间，赢得了业务！

这就是渐进投资软件开发的起源，此后诞生了本书所要介绍的渐进投资方法(IFM，Incremental Funding Methodology)。自然，这个方法要用几个月的时间才能产生效果。任何观点的酝酿都是一个不可预知的过程，在它产生作用之前谁都不清楚会发生什么。虽然渐进投资软件开发的观点已经初步显示出了一些成功的迹象，但是还需要更全面地加以证实。1999 年末我受命负责 Sun Microsystems 公司的纽约“Java 中心”。这个 Java 中心是 Sun 公司的全球 Java 咨询机构的一个实践，为客户提供体系结构和设计方面的的专业知识来解决在 Java 和 J2EE 中遇到的问题。很快我们就会看到，通过将最初的 IFM 观念写成应用软件开发工作的提议，就能够获得并成功交付几宗大合同，特别是有关金融产业的用户。然而，我们了解到渐进的投资和早期的价值发布严格依赖于需求本质和价值优化的重分配能力。需求工程本身就是一门学科。

本书的宗旨是希望它能给开发人员、经理、商业主管和风险投资者的成功提供一定的帮助。如果它提供了一个所有开发机构都能用来交换观点的通用的专业词汇表，这些观点是以提高经济上的成功率，降低应用程序

此为试读，需要完整PDF请访问：www.ertongbook.com

开发项目的风险为目的的，那么它就达到了它最主要的目标。可以进一步帮助读者的是，本书有一个相关网站 http://www.softwarebynumbers.org，在这个网站上您可以得到更多的信息和可下载的工具来帮助您取得进步。

我们相信您会发现本书的内容富有挑战性，能让读者轻松愉快地学习。我们欢迎您通过网站给我们发来反馈信息。

祝愿您所有的软件项目都取得成功，并且有利可图。

Mark Denne

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第1章

dot.com 之后的软件开发

当今，既然 dot.com 的幻想已经破灭，那么软件开发就需要新的和大胆的手段来应对投资周期更短，市场反应更快，以及运作灵活性更强的商业需求。现在应该打开技术驱动的开发方法的黑匣子，同时应该认识到软件创造在本质上是一个价值创造的过程。现在应该使软件开发服从于详细的金融审查和责任机制，这一机制也是其他任何一个价值创造过程所要服从的机制。

1.1 软件开发并不简单

“软件开发是简单的”，这是 IDE(Integrated Development Environment，集成开发环境)的制造商在 21 世纪初提出的大胆宣言。只要学会语言，掌握工具，使用我们的 IDE，您就能制作出高质量的软件。您将会在财政预算之内准时交货，同时实现软件的自文档化。所开发出来的软件会在 Internet 时代竞争激烈的世界中最先上市，而且它将创造金钱，很多的金钱。

遗憾的是，情况并非上面所说的那样。

我们可以了解到无数关于项目取消或失败的报道，这些报道给我们发出了一个清晰的信息：软件开发会是一项昂贵而又有风险的尝试。2000 年和 2001 年的

那场“dot 炸弹”给投资所带来的影响仍然令风险投资者不寒而栗。调查机构估算了失败软件项目的成本，仅仅在美国，平均每年的损失就达到了惊人的11位数美元。系统集成人员力争用最大数量的商用现货供应(COTS)软件为他们的客户提供集成系统，因为降低一个应用软件开发项目的成本和风险的最有效方法就是避免完全由自己编写软件。

为了说明软件应用开发放弃业界流行做法达到了怎样的程度，下面举一个例子。一个财富500强的公司最近针对所有应用程序开发项目设立了一个最高限度——10万美元(在商界，10万美元对于应用软件开发而言是一个极低的指标)。如有例外需得到CIO和CEO的批准。这个例子表明：我们难以找到一个更有力的证据来说明我们所预知的风险达到了一个什么样的程度。

那么，究竟是哪里出了问题呢？我们已经有了五十年高级语言开发的经验，具有面向对象软件的可重用性优点、丰富的开发方法、组件化的应用程序、迭代式的设计技术，和非常成熟的集成开发环境。还有，由于桌面计算机的能力成指数增长，我们能比从前更快地进行编写-编译-测试的循环过程。我们在学校里教学生们信息技术，在大学中艰辛地培养出计算机科学的毕业生，流行的编程语言(如Java)的已注册开发人员数以百万计。那么，为什么我们还不能成功地编写软件呢？

20世纪末的社会学家和媒体专家越来越关注关系之间的脱节和社会经济脱节的话题。在他们进行这些方面的研究时，所采用的一个普遍途径是，将信息时代无处不在的连通要求与西方社会不同阶层之间所能感受到的迅速显现的断裂感和疏远感相提并论。我们所面对的事实是：软件开发的教育与实践已经日益脱节，在语言和价值两个方面都存在这种现象，有人根据这两个方面来定义需求，并明确说明该软件的要求。应用软件开发人员以方法、类和用例与别人交流，而客户和业主关心的则是适销特性、资金流量和投资回报率(ROI)。这两个世界之间的联系点非常有限。以最近对有关软件体系结构和面向对象设计的16本书所做的分析和调查为例，结果表明，只有两本书在索引中包含“成本”一词。

在业务交往中，对于IT部门的看法进一步证明了这两个世界之间的分离。虽然IT部门是当今时代的企业中最具代表性的最大成本中心之一，但

是我们必须提醒自己——仅仅在 40 年前 IT 部门还并不存在。在会计领域里普遍的感觉是，IT 部门是一个不盈利的部门(不存在价值创造)，而且由于 IT 技术通常不是核心技术，甚至与大部分商贸活动没有关系。因此，在 20 世纪 90 年代，IT 部门随时可能被外包的情况就不足为奇了。

正如 Barry Boehm 所认识到的那样，真正有开导意义的观点是：软件开发不是一种由于外购而造成的成本，而应该是一种“创造价值的活动”。遗憾的是，软件工程师通常不卷入进来，或者通常不理解企业层次上的价值创造目标。

新西兰 boutique 软件开发公司 Green Door 服务中心得出一个结论，并将其简单归结为“软件开发是简单的，但听信它并理解它却是困难的”。

1.2 从新的视角考虑软件开发

软件开发过程和软件工程并没有失去方法上的支持。事实上，在面向对象编程领域中的开发人员几乎都注意到了方法的运用，如 RUP(Rational Unified Process，Rational 统一过程)或 XP(eXtreme Programming，极限编程)。其他方法包括 Peter Coad 的 FDD(Feature-Driven Development，特性驱动开发)，Sun Microsystems 的 SunTone AM(SunTone Architecture Methodology，SunTone 体系结构方法)。因此，在已经存在这么多方法的情况下，再提出另一种方法似乎没有必要。

本书没有打算提出一种新方法，而是在迭代开发方法的基础上进行相关的分析。本书介绍了一些过程和指导方针，通过这些过程和指导方针，您可以认识到软件开发是一个创造价值的过程，这个过程主要受成本和投资因素的限制，对于这个过程来说，最重要的目标就是金融风险的消除或缓解。这实际上是一个整体性的方法，在此方法中客户的需要和开发人员的需求同等重要，并且客户和开发人员的参与程度是相同的。这种在价值创造环境中建立起来的迭代开发的方法是本书要阐述的重点内容。

1.3 吸取历史教训

20 世纪 90 年代末，出现了以 dot.com 的高风险投资为基础的商业主张，而这些主张存在严重的缺陷。过去几年，人们一直对这些主张进行大量的详细审查。在当时那个时期，投资者和软件开发人员很少关注传统的 ROI 思想。那时最流行的看法是，投资于软件业的投资者是通过公司在市场上增加的资本价值(即未来的销售额和利润的预期值)来获得回报的，而不是通过更好的方式(如每股盈利)来获得回报。

如果您认为传统的 ROI 思想现在仍旧行得通，那么您的这种想法是可以原谅的。然而，您的这种想法并不符合实际情况。事实是，传统 ROI 的特定概念本身面临详细审查。现在很多机构不会容忍超过一年的 ROI。而当我们想到一个三到五年的 ROI 曾是(前面所述的)dot.com 繁荣兴旺的那个时代的标准，我们就会感到吃惊。

这是不是意味着传统 ROI 就没有一点用处了呢？也许。但是，只要这种思想持续下去，它就会为软件开发人员和提供软件开发服务的机构带来一些独特的问题。这是因为，对于一个大的软件开发项目而言，几乎不可能在短于一年的时间内收回投资。所以，如果短期 ROI 是惟一可接受的 ROI，而典型的软件开发则要求更长时间的 ROI，那么如何在后 dot 爆炸的时代进行软件开发的投资呢？如何才能将必要的资金投入大的软件开发项目上而仍旧使开发人员和投资者都能赚到钱呢？

答案在于对软件的传统 ROI 模型进行重新评估，并掌握软件设计方法中进行短期开发的基本原理。

在过去的大约十年时间内，软件开发的方法论方案无论在本质上还是在手段上都有重大的变化。随着这些思想变革的广泛传播，通过提供大量详细的功能规范和开发一系列谨慎调整的小模块(以后再对这些小模块进行组装、集成与测试)来开发软件的这种思想，在很大程度上已经被人们摒弃掉了。这其中有许多原因。

对于“瀑布”方案的问题，简单地说，是它没有考虑到商业的现实性，

此为试读，需要完整PDF请访问：www.ertongbook.com