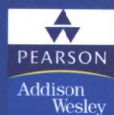


卡内基·梅隆大学软件工程丛书



遗留系统的现代化改造

—— 软件技术、工程过程和业务实践

Modernizing Legacy Systems:

(影印版)

Software Technologies, Engineering Processes, and Business Practices

罗伯特·C. 塞克德 [Robert C. Seacord]
(美) 丹尼尔·普拉考士 [Daniel Plakosh]
格雷丝·A. 刘易斯 [Grace A. Lewis]

著



清华大学出版社



卡内基·梅隆大学软件工程丛书

遗留系统的现代化改造

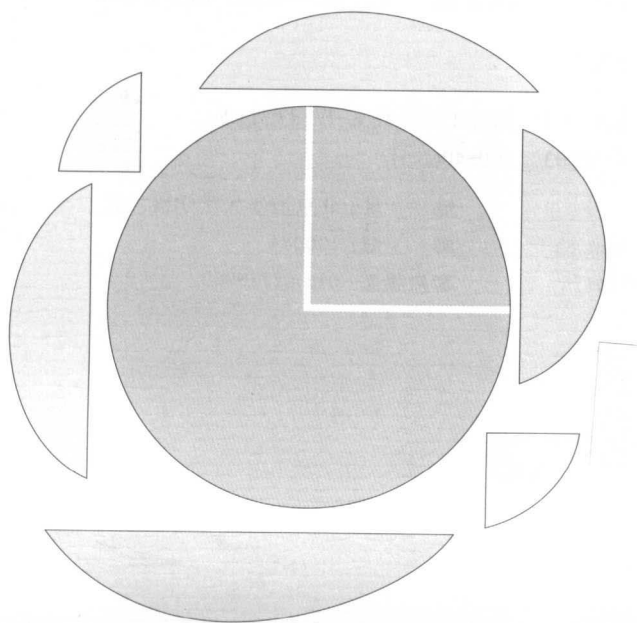
—— 软件技术、工程过程和业务实践

Modernizing Legacy Systems:

(影印版)

Software Technologies, Engineering Processes, and Business Practices

罗伯特·C. 塞克德 [Robert C. Seacord]
(美) 丹尼尔·普拉考士 [Daniel Plakosh] 著
格雷斯·A. 刘易斯 [Grace A. Lewis]



清华大学出版社
北京

English reprint edition copyright © 2003 by PEARSON EDUCATION ASIA
LIMITED and TSINGHUA UNIVERSITY PRESS.

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

Original English language title: Modernizing Legacy Systems: Software Technologies,
Engineering Processes, and Business Practices, 1st Edition, by Robert C. Seacord,
Daniel Plakosh, Grace A. Lewis, Copyright © 2003

All Rights Reserved.

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

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

本书影印版由 Pearson Education 授权给清华大学出版社出版发行。

**For sale and distribution in the People's Republic of China
exclusively (except Taiwan, Hong Kong SAR and Macao SAR).**
仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国
台湾地区)销售发行。

北京市版权局著作权合同登记号 图字: 01-2003-5354

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

遗留系统的现代化改造——软件技术、工程过程和业务实践=Modernizing Legacy
Systems: Software Technologies, Engineering Processes, and Business Practices /
(美)塞克德,(美)普拉考士,(美)刘易斯著. —影印本. —北京:清华大学出
版社, 2003

(卡内基·梅隆大学软件工程丛书)

ISBN 7-302-07272-8

I. 遗… II. ①塞… ②普… ③刘… III. 软件工程—英文 IV. TP311.5

中国版本图书馆 CIP 数据核字(2003)第 084066 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

文稿编辑: 刘伟琴

封面设计: 立日新设计公司

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 148×210 印张: 11.5 插页: 1

版 次: 2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

书 号: ISBN 7-302-07272-8/TP·5279

印 数: 1~3000

定 价: 29.00 元

出版说明

1984年,美国国防部出资在卡内基·梅隆大学设立软件 Engineering Institute, 简称 SEI)。SEI 于 1986 年开始研究软件过程能力成熟度模型 (Capability Maturity Model, CMM), 1991 年正式推出了 CMM 1.0 版, 1993 年推出 CMM 1.1 版。此后, SEI 还完成了能力成熟度模型集成 (Capability Maturity Model Integration, 简称 CMMI)。目前, CMM 2.0 版已经推出。

CMM 自问世以来备受关注, 在一些发达国家和地区得到了广泛应用, 成为衡量软件公司软件开发管理水平的重要参考因素, 并成为软件过程改进的事实标准。CMM 目前代表着软件发展的一种思路, 一种提高软件过程能力的途径。它为软件行业的发展提供了一个良好的框架, 是软件过程能力提高的有用工具。

SEI 十几年的研究过程和成果, 都浓缩在由 SEI 参与研究工作的资深专家亲自撰写的卡内基·梅隆大学软件工程丛书 (SEI Series In Software Engineering) 中。

为增强我国软件企业的竞争力, 提高国产软件的水平, 清华大学出版社全面引进了这套丛书, 分批影印和翻译出版, 这套丛书采取开放式出版。不断改进, 不断出版, 旨在满足国内软件界人士学习原版软件工程高级教程的愿望。

清华大学出版社

卡内基·梅隆大学软件工程丛书

编 委 会 名 单

主 任 周伯生

副 主 任 郑人杰

委 员 (按姓名拼音顺序排列)

董士海 顾毓清 王 纬

吴超英 尤晓东

执行委员 尤晓东

秘 书 廖彬山

总 序

周伯生

美国卡内基·梅隆大学软件工程研究所（CMU/SEI）是美国联邦政府资助构建的研究单位，由美国国防部主管。他们确认，为了保证软件开发工作的成功，由软件开发人员、软件采办人员和软件用户组成的集成化团队必须具有必要的软件工程知识和技能，以保证能按时向用户交付正确的软件。所谓“正确的”就是指在功能、性能和成本几个方面都能满足用户要求且无缺陷；所谓“无缺陷”就是指在编码后对软件系统进行了彻底的穷举测试修复了所有的缺陷，或保证所编写的代码本身不存在缺陷。

CMU/SEI 为了达到这个目的，提出了创造、应用和推广的战略。这里的“创造”是指与软件工程研究社团一起，共同创造新的实践或改进原有的实践，而不墨守成规。这里的“应用”是指与一线开发人员共同工作，以应用、改进和确认这些新的或改进的实践，强调理论联系实际。这里的“推广”是指与整个社团一起，共同鼓励和支持这些经过验证和确认的、新的或改进的实践在世界范围内的应用，通过实践进行进一步的检验和提高。如此循环，往复无穷。

他们把所获得的成就归纳为两个主要领域。一个是倡导软件工程管理的实践，使软件组织在采办、构建和改进软件系统时，具有预测的能力与控制质量、进度、成本、开发周期和生产效率的能力。另一个是改进软件工程技术的实践，使软件工程师具有分析、预测和控制软件系统属性的能力，其中包括在采办、构建和改进软件系统时，能进行恰当的权衡，作出正确的判断和决策。

CMU/SEI 通过出版软件工程丛书, 总结他们的研究成果和实践经验, 是推广这两个领域经验的重大举措。

卡内基·梅隆大学软件工程丛书由 CMU/SEI 和 Addison-Wesley 公司共同组织出版, 共分 4 个部分: 计算机和网络安全 (已出版了 2 本著作), 工程实践 (已出版了 8 本著作), 过程改进和过程管理 (已出版了 11 本著作), 团队软件过程和个体软件过程 (已出版了 3 本著作)。前两者属于软件工程技术实践, 后两者属于软件工程管理实践。目前这 4 个部分共出版了 24 本著作, 以向软件工程实践人员和学生方便地提供最新的软件工程信息。这些著作凝聚了全世界软件工程界上百位开拓者和成千上万实践者的创造性劳动, 蕴含了大量的宝贵经验和沉痛教训, 很值得我们学习。

清华大学出版社邀请我和郑人杰教授共同组织卡内基·梅隆大学软件工程丛书编委会。清华社计划首先影印 6 本著作, 翻译出版 15 本著作。据我所知, 在 Addison-Wesley 公司出版的 SEI 软件工程丛书中, 人民邮电出版社已经翻译出版了《个体软件过程》和《团队软件过程》, 还拟影印出版《个体软件过程》和《软件工程规范》; 电子工业出版社已经翻译出版了《净室软件工程的技术与过程》、《能力成熟度模型 CMM 1.1 指南》、《能力成熟度模型集成 CMMI》和《软件项目管理》; 北京航空航天大学出版社已经翻译出版了《统计过程控制》。这些出版社共计影印 2 本著作, 翻译出版 7 本著作。这样, 可以预期我国在今年年底共可影印 8 本著作, 翻译出版 22 本著作。各个出版社的有远见的辛勤劳动, 为我们创造了“引进、消化、吸收、创新”的机遇。我们应该结合各自的实践, 认真学习国外的先进经验, 以大大提高我国软件工程的理论和实践水平。

在这套丛书中, 特别值得一提的是, 在过程工程领域被誉为软件过程之父的 Humphrey 先生所撰写的《软件过程管理》、《技

术人员管理》、《软件工程规范》、《个体软件过程》、《团队软件过程》和《软件制胜之道》等 6 本著作，将于今年年内全部翻译出版，其中《软件过程管理》、《技术人员管理》、《软件工程规范》、《个体软件过程》和《软件制胜之道》等 5 本著作亦已经或将于今年年内影印出版。

《软件过程管理》是软件过程领域的开创性著作，是为软件公司经理和软件项目经理撰写的。用这本书提出的原理来指导软件开发，可以有效地按照预定进度得到高质量的软件，同时还可了解如何持续进行过程改进。美国 CMU/SEI 按照这本书提出的原理开发了能力成熟度模型，在国际上得到绝大多数国家的认可和广泛采用，是改进软件过程能力的有力武器。在信息技术迅速发展和企业激烈竞争的今天，能否持续改进过程往往决定企业的命运。

作为一个软件经理，在改进组织的能力之前，首先必须明确绝大多数软件问题是由管理不善所引起的。因此，要改进组织的性能，首先需要改进自己的管理模式。同时还要认识到软件开发是一项智力劳动，需要拥有掌握高技能和忘我工作的技术人员。因此，有效的软件管理需要充分注意技术人员的管理。

《技术人员管理》这本著作就是为达到这个目的而撰写的。高质量的技术工作要求没有差错，这就要求人们高度专心和高度献身。因此要求人们对他所从事的工作不仅具有高度的责任感，而且具有浓厚的兴趣和高度的热忱。在当前知识经济群龙相争的今天，一个能激励人们进行创造性工作的领导群体，是众多竞争因素中最重要的因素。本书提供了大量的实用指南，可用来有效地改进工程人员、经理和组织的性能。

Humphrey 先生还认为这本书特别适合于在我国工作的软件经理。我国是一个人口大国，拥有大量能干的知识分子，而且信息领域的劳动力价格比国际市场的价格要低，因此吸引了许多国

家到我国来投资。但若不提高人员的素质,不在产品质量和进度方面也狠下功夫,就不能在这方面持续保持优势。

《软件工程规范》是为编程人员撰写的。它精辟地阐述了个体软件过程(PSP)的基本原理,详尽地描述了人们如何来控制自己的工作,如何与管理方协商各项安排。在软件工程界,这本著作被誉为是软件工程由定性进入定量的标志。目前在世界范围内,有成千上万的软件工程技术人员在接受有关PSP的培训,以便正确地遵循PSP的实践、开发和管理工作计划,在他们承诺的进度范围内,交付高质量的产品。

《软件制胜之道》这本著作描述了团队软件过程的基本原理,详尽地阐述了在软件组织中如何应用PSP和TSP的原理以及它所能带来的效益。此外,虽然CMM同样适用于小型组织,但在其他著作中都没有描述如何应用CMM于个体或小型团队,这本书填补了这个空白。应该指出,如果一个组织正在按照CMM改进过程,则PSP和TSP是和CMM完全相容的。如果一个组织还没有按照CMM改进过程,则有关PSP和TSP的训练,可以为未来的CMM实践奠定坚实的基础。

在软件工程技术实践方面目前共出版了10本著作,其中《用商业组件构建系统》、《软件构架实践》和《软件构架评估——方法和案例研究》等3本著作详尽地阐述了软件构架的构建、实践和评估。鉴于是否有一个稳定的软件构架,对软件的质量和成本影响很大,因此如何获得一个好的构架就成为当今软件界研究的重点。我相信这几本著作的出版,将对我国软件构架领域的研究与实践有重要的参考价值。此外,众所周知,计算机与网络的安全问题对信息系统的可靠使用关系极大,《CERT安全指南》的出版将会对我国在这一领域的研究和实践起积极的促进作用。《风险管理——软件系统开发方法》、《软件采办管理——开放系统和COTS产品》、《项目管理原理》、《软件产品线——实践与模式》

和《系统工程——基于信息的设计方法》等 5 本著作，分别从风险管理、软件采办、项目管理、软件产品线以及信息系统设计方法等几个方面阐述了大型、复杂软件系统的开发问题，是有关发展软件产业的重要领域，很值得我国软件产业界借鉴。

目前我们所处的时代是信息化时代，是人类进入能够综合利用物质、能量和信息三种资源的时代。千百年来以传统的物质产品的生产、流通、消费为基本特征的物质型经济，将逐步进入以信息产品的生产、流通、利用和消费为基本特征的知识型经济。在这个历史任务中，建造和广泛应用各类计算机应用系统是其公共特征。计算机软件是计算机应用系统的灵魂，没有先进的软件产业，不可能有先进的信息产业，从而也不可能建成现代化的知识型经济。

我们应该看到，在软件领域中我国在总体上离世界先进水平还有相当大的差距。但是，我们不能跟随他国的脚印，走他人的老路。我们应该抓住机遇，直接针对未来的目标，在软件工程技术和软件工程管理两个方面，注意研究卡内基·梅隆大学软件工程丛书中倡导的原理和方法，联系实际，认真实践，并充分利用我国丰富优秀的人力资源和尊重教育的优良传统，大力培养各个层次的高质量的软件工程人员，使其具有开发各类大型、复杂软件系统的能力。我衷心地预祝清华大学出版社影印和翻译出版这套丛书，在把我国建设成为一个真正现代化的软件产业大国的历史任务中起到推波助澜的作用，并请读者在阅读这些译著时，对这套丛书的选题、译文和编排等方面都提出批评和建议。

周伯生

于北京

2002 年 8 月 18 日

致 谢

在认识我们的遗留系统的精神中，我谨以此书纪念我的祖父母——Charles, Ruth, Andrew 和 Flora；奉献给我的父母 Charles 和 Adrienne；奉献给我的妻子 Rhonda 和我们的孩子 Chelsea 和 Jordan。

——rCs

我谨以此书奉献给我的妻子 Judy，感谢她的爱和支持；还要奉献给我的父母 Charles 和 Dolores，感谢他们给予对我的教育。

——Dan

我谨以此书奉献给那些使我的生命变得精彩的人：我的丈夫 Mike，感谢他的爱和支持；奉献给我的“妈咪”Graciela，感谢她把我引导到我今天所处的位置；还要奉献给我的姐姐 Ingrid，感谢她总是陪伴在我的身边。

——Grace

前 言

当软件系统难以修改和演化的时候，就变成了遗留系统。但是，包含在遗留系统中的知识构成了相当重要的企业资源。假设这些系统仍然能提供相当重要的业务价值，那么就必须对它们进行现代化改造或者替换。本书描述了一种遗留系统现代化改造的风险管理方法，它应用了软件技术的知识和对业务环境中工程过程的理解。

面向的读者

本书对于参与现代化改造遗留系统的所有人来说都应该是非常有用的。

- 对于软件工程师，本书将帮助你理解推动现代化改造工作的某些更大的业务关注点。
- 对于软件设计师，本书将帮助你在设计过程中，理解遗留代码的影响以及增量的开发和部署实践。
- 对于系统构架师，本书解释了那些在实践中失败或者成功的过程和技术。它提供了对你如何重复这些成功和避免那些失败的深刻见解。
- 对于 IT 经理，本书解释了技术和业务目标是如何影响软件现代化改造过程的。

特别要指出的是，本书可以帮助读者回答下面几个问题：

- 何时决定和如何决定现代化改造工作或者替换工作是应当的呢？
- 怎么建立对遗留系统的理解呢？
- 如何理解用于对系统进行现代化改造的信息系统技术，以及如何评估这些技术的适用性呢？
- 什么时候与涉众交涉，以及怎样协调他们的互相矛盾的需求呢？
- 构架在遗留系统的现代化改造中扮演什么样的角色呢？
- 怎样估计一个遗留系统现代化改造的成本呢？
- 怎样评估和选择一个现代化改造策略呢？
- 怎样制定一个详细的现代化改造计划呢？

本书的组织 and 内容

本书展示了怎样增量地对遗留系统进行现代化改造。它使用和扩展了在《用商业组件构建系统》（由清华大学出版社出版）一书中描述的方法和技术，尽早在概念阶段使用工程专业知识，以确保制定现实的和全面的计划。

本书特别描述了一个广泛的案例研究，涉及一项重要的现代化改造工作。这个案例研究中的遗留系统包含近 200 万行 COBOL 代码，它们已经被开发出来 30 多年了。人们正在用一个基于 Java2 企业版（J2EE）构架的现代系统来替换它。其他的挑战包括一个要求增量开发和部署系统的需求。我们着眼于：用来现代化改造此系统的策略；企业 JavaBeans、面向消息的中间件、Java 和其他 J2EE 技术的使用，来产生现代化系统；支持软件工程过程和技术；以及得到的系统。

第 1 章介绍了软件演化的挑战和实践。第 2 章介绍了本书中的主要案例研究。第 3 章介绍了风险管理的现代化改造 (RMM) 方法, 第 4~17 章详细阐述了这种方法, 并且通过案例研究对此作了解释。在第 4~17 章的开头, 我们提供了一个 RMM 活动图, 作为此章的一个路线图。第 18 章提出了一些建议, 帮助指导现代化改造工作, 但是, 如果不阅读本书的主要部分, 就不可能完全理解这些建议。

在整本书中, 我们使用统一建模语言 (Unified Modeling Language, UML) 表示构架绘图和设计模式。第 6 章简要介绍了 UML。

要了解与本书有关的最新信息、事件和新闻, 请访问 <http://www.sei.cmu.edu/cbs/mls>。

Preface

Software systems become legacy systems when they begin to resist modification and evolution. However, the knowledge embodied in legacy systems constitutes a significant corporate asset. Assuming that these systems still provide significant business value, they must then be modernized or replaced. This book describes a risk-managed approach to legacy system modernization that applies a knowledge of software technologies and an understanding of engineering processes within a business context.

Audience

Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices should be useful to anyone involved in modernizing a legacy system.

- For a *software engineer*, the book should help you understand some of the larger business concerns that drive a modernization effort.
- For a *software designer*, this book should help you understand the impact of legacy code, coupled with incremental development and deployment practices, on design activities.
- For a *system architect*, this book explains the processes and techniques that have failed or succeeded in practice. It should also provide insight into how you can repeat these successes and avoid the failures.
- For an *IT manager*, this book explains how technology and business objectives influence the software modernization processes.

In particular, the book should help you answer the following questions:

- When and how do I decide whether a modernization or replacement effort is justified?
- How do I develop an understanding of the legacy system?
- How do I gain an understanding of, and evaluate the applicability of, information system technologies that can be used in the modernization of my system?

- When do I involve the stakeholders, and how can I reconcile their conflicting needs?
- What role does architecture play in legacy system modernization?
- How can I estimate the cost of a legacy system modernization?
- How can I evaluate and select a modernization strategy?
- How can I develop a detailed modernization plan?

Organization and Content

Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices shows how legacy systems can be incrementally modernized. It uses and extends the methods and techniques described in *Building Systems from Commercial Components* [Wallnau 01] to draw on engineering expertise early in the conceptual phase to ensure realistic and comprehensive planning.

This book features an extensive case study involving a major modernization effort. The legacy system in this case study consists of nearly 2 million lines of COBOL code developed over 30 years. The system is being replaced with a modern system based on the Java 2 Enterprise Edition (J2EE) architecture. Additional challenges include a requirement to incrementally develop and deploy the system. We look at the strategy used to modernize the system; the use of Enterprise JavaBeans, message-oriented middleware, Java, and other J2EE technologies to produce the modern system; the supporting software engineering processes and techniques; and the resulting system.

Chapter 1 provides an introduction to the challenges and practices of software evolution. Chapter 2 introduces the major case study in the book. Chapter 3 introduces the risk-managed modernization (RMM) approach, which is elaborated in Chapters 4 through 17 and illustrated by the case study. At the beginning of Chapters 4 through 17, we provide an activity diagram of RMM as a road map to the chapter. Chapter 18 provides some recommendations to help guide your modernization efforts, although these recommendations cannot be fully appreciated without reading the main body of the book.

Throughout this book, we use the Unified Modeling Language (UML) to represent architecture drawings and design patterns. A brief introduction to UML is provided in Chapter 6.

Updated information, events, and news related to *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices* can be found at <http://www.sei.cmu.edu/cbs/mls>.

Acknowledgments

First, we want to thank Santiago Comella-Dorda, Vivian Martin, Len Bass, Felix Bachmann, Paul Clements, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford, Lutz Wrage, Russ Bunting, Dennis Smith, Liam O'Brien, Scott Tilley, David Zubrow, Brad Clark, and Wolf Goethert for their contribution of chapters in this book. Many people also contributed material either directly or indirectly, including John Robert, Patrick Place, Scott Hissam, Kurt Wallnau, Mark McLaughlin, and Robert Graham. We also owe a debt of gratitude to our manager, John Foreman, for his strong support, without which we would not have succeeded; and also the strong support we received from Scott Vesper, Lester Reagan, and Jon Dittmer.

Special thanks go to our in-house editors, Len Estrin and Pennie Walters, for their efforts in making our writing appear acceptable, Barbara White and Suzanne Couturiaux for making our work presentable, Shelia Rosenthal for helping us with the bibliography, Peter Gordon from Addison-Wesley for keeping the effort on target, and the Addison-Wesley production team, including Elizabeth Ryan, John Fuller, and Karin Hansen, and Evelyn Pyle and Rob Mauhar.

We are also grateful to the reviewers, whose insightful comments are reflected throughout our work: Thomas Soller, Ed Morris, Steven Seacord, Edward Neubecker, and Peter Bye. We also happily acknowledge the contributions of members not already mentioned of the Software Engineering Institute (SEI) COTS-Based Systems initiative. Last, we are indebted to Steve Cross, the director of the SEI, for providing an environment in which software engineering theory can be hardened in practice and promoted in public.