



SEI 软件工程丛书·影印版

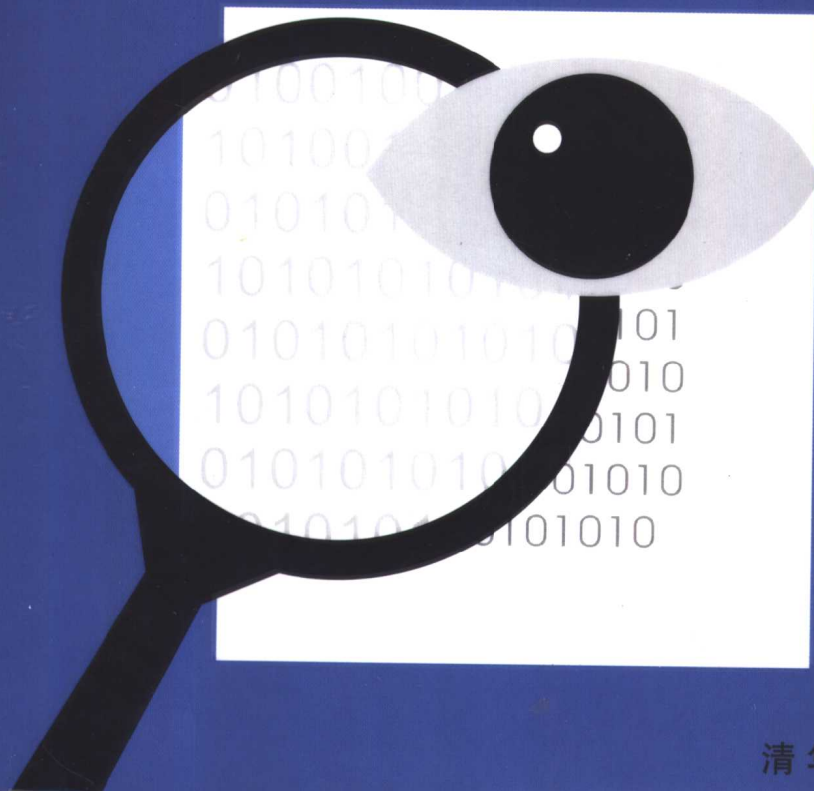
卡内基·梅隆大学软件工程研究所推荐·软件学院教材

PEARSON
Addison
Wesley

软件构架评估

Evaluating Software Architectures

保罗·克莱门茨 [Paul Clements]
[美] 瑞克·凯兹曼 [Rick Kazman] 著
马克·克莱恩 [Mark Klein]



清华大学出版社



SEI 软件工程丛书·影印版

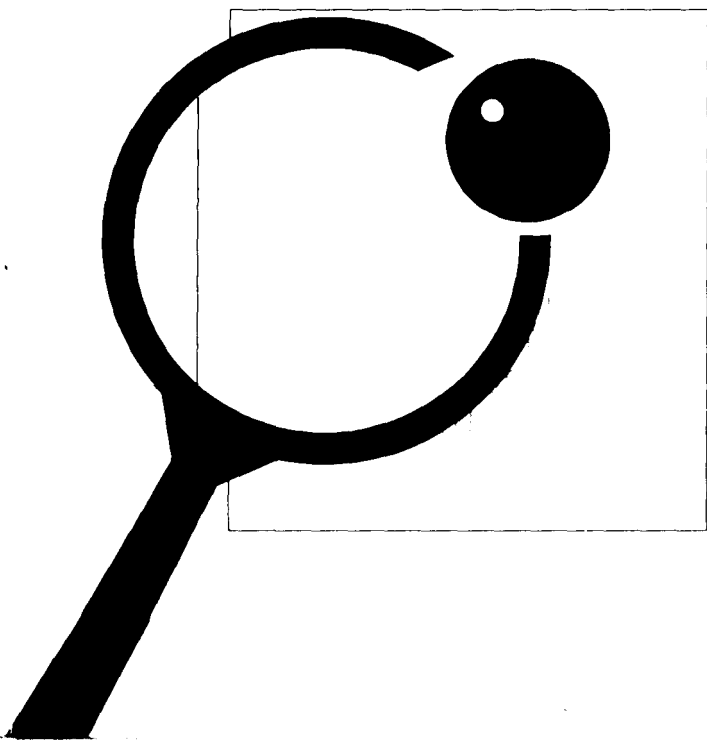
卡内基·梅隆大学软件工程研究所推荐·软件学院教材

软件构架评估

保罗·克莱门茨 [Paul Clements]

[美] 瑞克·凯兹曼 [Rick Kazman] 著

马克·克莱恩 [Mark Klein]



清华大学出版社
北京

Evaluating Software Architectures

Methods and Case Studies

Paul Clements

Rick Kazman

Mark Klein

◆ Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal

London • Munich • Paris • Madrid

Capetown • Sydney • Tokyo • Singapore • Mexico City

内 容 简 介

本书对大型软件系统开发中具有重要意义的软件构架评估进行了系统阐述,着重讲解了3种流行的构架评估方法。本书是构架评估的实践指南,给出了评估方法的应用案例和细节,使读者在读完本书后即可应用这些方法进行构架评估。

本书是卡内基·梅隆大学软件工程研究所推荐教材,适用于软件构架评估人员,也可作为软件学院及大学计算机等专业相关课程的教材,还可作为软件公司各级管理和开发人员参考。

EISBN: 0-201-70482-X

Evaluating Software Architectures: Methods and Case Studies.

Paul Clements, Rick Kazman, Mark Klein

Copyright © 2002 by Addison-Wesley

Original English language edition published by Addison-Wesley.

All right reserved.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau 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。

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。

北京市版权局著作权合同登记号:图字 01-2003-0863 号

图书在版编目(CIP)数据

软件构架评估 Evaluating Software Architectures / (美)克莱门茨, (美)凯兹曼, (美)克莱恩著. 影印版. —北京:清华大学出版社, 2003

(SEI 软件工程丛书)

ISBN 7-302-06389-3

I. 软... II. (1)克... (2)凯... (3)克... III. 软件开发—评估—英文 IV. TP311.52

中国版本图书馆CIP数据核字(2003)第018762号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.com.cn>

<http://www.tup.tsinghua.edu.cn>

责任编辑: 尤晓东

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 23.75 插页: 1

版 次: 2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

书 号: ISBN 7-302-06389-3 TP·4816

印 数: 0001—3000

定 价: 49.00 元



Carnegie Mellon Software Engineering Institute

The SEI Series in Software Engineering

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley, Inc. was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

CMM, Capability Maturity Model, Capability Maturity Modeling, Carnegie Mellon, CERT, and CERT Coordination Center are registered in the U.S. Patent and Trademark Office.

ATAM; Architecture Tradeoff Analysis Method; CMMI; CMM Integration; CURE; IDEAL; Interim Profile; OCTAVE; Operationally Critical Threat, Asset, and Vulnerability Evaluation; Personal Software Process; PSP; SCAMPI; SCAMPI Lead Assessor; SCE; Team Software Process; and TSP are service marks of Carnegie Mellon University.

ANY MATERIAL FURNISHED BY CARNEGIE MELLON UNIVERSITY AND THE SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for special sales. For more information, please contact:

Pearson Education Corporate Sales Division
201 W. 103rd Street
Indianapolis, IN 46290
(800) 428-5331
corpsales@pearsoned.com

Visit AW on the Web: www.awl.com/cseng/

Library of Congress Cataloging-in-Publication Data

Clements, Paul, 1955—

Evaluating software architectures : methods and case studies / Paul Clements, Rick Kazman, Mark Klein.

p. cm. — (SEI series in software engineering)

Includes bibliographical references and index.

ISBN 0-201-70482-X

1. Computer software—Evaluation. 2. Computer architecture—Evaluation. I. Kazman, Rick. II. Klein, Mark. III. Tide. IV. Series.

QA76.754.C47 2002
005.1—dc2

2001045880

Copyright © 2002 by Addison-Wesley

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 the publisher. Printed in the United States of America. Published simultaneously in Canada.

0-201-70482-X

Text printed on recycled paper

1 2 3 4 5 6 7 8 9 10—CRW—0504030201

First printing, October 2001

出版说明

1984 年,美国国防部出资在卡内基·梅隆大学设立软件工程研究所(Software 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 SEI 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, 经清华大学出版社和三联四方工作室共同策划, 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

2002 年 8 月

编 委 会

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。

SEI 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 SEI 软件工程译丛编委会。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員正在接受有关 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

一个良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——系统与网络安全实践》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——实践和模式》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 SEI 软件工程丛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

Praise for *Evaluating Software Architectures*

"The architecture of complex software or systems is a collection of hard decisions that are very expensive to change. Successful product development and evolution depend on making the right architectural choices. Can you afford not to identify and not to evaluate these choices? The authors of this book are experts in software architecture and its evaluation. They collected a wealth of ideas and experience in a well-organized and accessible form. If you are involved in the development of complex systems or software, you will find this book an invaluable guide for establishing and improving architecture evaluation practice in your organization."

—Alexander Ran, Principal Scientist of Software Architecture, Nokia

"Software engineers must own this book. It is a well-written guide to the steps for evaluating software architecture. It argues for the inclusion of architecture evaluation and review as a standard part of the software development lifecycle. It introduces some new and innovative methods for analyzing important architecture characteristics, like extensibility, portability, and reliability. I believe these methods will become new engineering cornerstones for creating good software systems."

—Joe Maranzano, AT&T Bell Labs Fellow in Software Architecture (1990),
and former head of the Bell Labs Software Technology Center

"Experience and teamwork are the only approaches I know of to deliver products faster, cheaper, and yet to delight your customers. In their first book, *Software Architecture in Practice*, Paul and Rick (and Len Bass) helped me match my experience with theory. Their invaluable approaches and case studies changed my practice and the way I proceed to design systems and software architectures. This second book, with Mark, covers what I will look at before I feel good about an architecture. It is about how I can tap other people's experience to produce an improved outcome, using other people's feedback. I have used many of the concepts explained in this book for my customers' benefit. Using this book, you—architects, developers, and managers—will develop a common language and practice to team up and deliver more successful products."

—Bertrand Salle, lead architect with a major telecommunications company

"If architecture is the foundation of system construction, architectural evaluation is part of the foundation of getting to a 'good' architecture. In this book, the authors put their considerable expertise to one of the most pressing issues in systems development today: how to evaluate an architecture prior to system construction to ascertain its feasibility and suitability to the system of interest. The book provides a practical guide to architecture evaluation using three contemporary evaluation methods. It should prove valuable to practitioners and as a basis for the evolution of architectural evaluation as an engineering practice."

—Rich Hilliard, Chief Technical Officer, ConsentCache, Inc.,
and technical editor, *IEEE Recommended Practice for
Architectural Description of Software-Intensive Systems*

"Too many systems have performance and other problems caused by an inappropriate architecture. Thus problems are introduced early, but are usually detected too late—when the deadline is near or, even worse, after the problem makes the headlines. Remedies lead to missed schedules, cost overruns, missed market windows, damaged customer relations, and many other difficulties. It is easy to prevent these problems by evaluating the architecture choices early, and selecting an appropriate one."

—Connie U. Smith, Ph.D., principal consultant, Performance
Engineering Services Division, L&S Computer Technology, Inc.,
and coauthor of the new book, *Performance Solutions:
A Practical Guide to Creating Responsive, Scalable Software*

"The ATAM [an evaluation method described in this book] is the natural quality-gate through which a high-level design should pass before a detail design project is initiated. Why use the ATAM to evaluate an architecture? Mitigation of design risk is a major reason, but more importantly, the ATAM provides an interactive vehicle that can give key development and user stakeholders architectural visibility—visibility that can lead to an important 'early buy-in.'"

—Rich Zebrowski, Software Technology Manager, Motorola, Inc.

"Caterpillar's experience with architecture reviews includes SAAM, ATAM, ARID, and ADR [evaluation methods described in this book, the first three in detail]. These reviews ensured that the needs of the user community were being met, and they exposed the architecture to others in the organization helping with understanding and organizational buy-in. The SAAM- and ATAM-based evaluations worked well to expose the architecture early in the development cycle to a broad range of people. The ARID- and ADR-based evaluations facilitated the exposure of technical details of the architecture later in the development cycle. As the architect of the pilot project for ARID, I observed that this review even served as an architecture training session before the architecture was fully documented."

—Lee R. DenBraber, former Lead Software Architect, Caterpillar, Inc.

“We’ve heard all the management hype about harnessing the innovative creativity of our teams, establishing integrated customer-developer-product teams, and better targeting our systems to meet end user needs. The ATAM techniques described in this book give technical managers, system architects, and engineers proven tools for breaking down the communications barriers that impede our ability to realize these goals. We have successfully integrated the ATAM techniques throughout our lifecycle, including development and maintenance, and have found that they provide the strong technical basis we need to evaluate the many difficult trades required by a system as large as EOSDIS.”

**—Mike Moore, Deputy Manager, Science Systems
Development Office, Earth Observing System Data
Information System (EOSDIS) Project,
NASA Goddard Space Flight Center**

“If you know how difficult architecture reviews are, you will be amazed how effective ATAM evaluations can be. For example, an ATAM evaluation we conducted on an important software product line identified a major architectural risk, which we subsequently were able to avoid—a benefit we expect to continue seeing. Moreover, ATAM techniques have enabled us to explain such risks to stakeholders far more clearly than by any other review method.”

—Stefan Ferber, Corporate Research, Robert Bosch GmbH

目 录

第 1 章 什么是软件构架	1
1.1 构架是风险承担者之间进行交流的手段	3
1.2 构架是最初设计决策的体现	11
1.3 构架是可重用、可传递的系统抽象	13
1.4 小结	14
1.5 进阶读物	15
1.6 讨论题	16
第 2 章 软件构架的评估	19
2.1 构架评估的原因	23
2.2 构架评估的时机	24
2.3 构架评估的参与者	26
2.4 构架评估的结果	27
2.5 构架评估所能考察的质量属性	30
2.6 模糊的质量属性不适于分析	32
2.7 构架评估的输出结果	34
2.8 构架评估的收益与成本	37
2.9 进阶读物	41
2.10 讨论题	42
第 3 章 ATAM 评估方法	43
3.1 ATAM 步骤概览	44
3.2 ATAM 评估步骤详述	45
3.3 ATAM 评估方法的阶段	70
3.4 进阶读物	84
3.5 讨论题	84
第 4 章 ATAM 案例研究（一）	87
4.1 准备工作	88
4.2 第 1 阶段	89
4.3 第 2 阶段	100
4.4 BCS 评估的结果	103
4.5 小结	107

4.6 讨论题	107
第 5 章 理解质量属性	109
5.1 质量属性刻画	110
5.2 在 ATAM 评估中应用质量属性刻画技巧	121
5.3 基于属性的构架样式	124
5.4 小结	125
5.5 进阶读物	126
5.6 讨论题	126
第 6 章 ATAM 案例研究 (二)	127
6.1 背景	128
6.2 第 0 阶段: 合作关系及准备工作	129
6.3 第 1 阶段: 初始评估	148
6.4 第 1 阶段和第 2 阶段之间的间隔	183
6.5 第 2 阶段: 评估的完成	184
6.6 第 3 阶段: 后续工作	202
6.7 进阶读物	209
6.8 讨论题	209
第 7 章 SAAM 评估方法	211
7.1 SAAM 概述	212
7.2 SAAM 评估的步骤	214
7.3 SAAM 评估日程示例	220
7.4 SAAM 案例研究	222
7.5 小结	238
7.6 进阶读物	239
7.7 讨论题	239
第 8 章 ARID 评估方法	241
8.1 积极设计评审	242
8.2 ARID: ADR 和 ATAM 的结合	245
8.3 ARID 方法的步骤	245
8.4 运用 ARID 方法的案例研究	248
8.5 小结	252
8.6 进阶读物	252
8.7 讨论题	253

第 9 章 软件构架评估方法比较	255
9.1 提问技巧	257
9.2 度量技巧	263
9.3 混合技巧	267
9.4 小结	271
9.5 进阶读物	273
9.6 讨论题	273
第 10 章 提升本组织的软件构架评估能力	275
10.1 取得编制上的批准	276
10.2 培养一批评估人员	276
10.3 建立本组织的知识库	278
10.4 小结	285
10.5 讨论题	285
第 11 章 结 语	287
11.1 已经就绪	287
11.2 本书所讲的评估方法	288
11.3 进行构架评估的原因	288
11.4 ATAM 方法能够起作用的原因	290
11.5 临别赠言	296
附录 A 基于质量属性的构架样式举例	297
A.1 问题描述	297
A.2 刺激 / 响应	298
A.3 构架样式	298
A.4 分析	299
A.5 进阶读物	302
参考文献	303
索引	307