

软件同级评审

Peer Reviews in Software A Practical Guide

(美) Karl E. Wiegiers 著

沈备军 宿为民 译



机械工业出版社
China Machine Press

TP311.5

57

软件同级评审

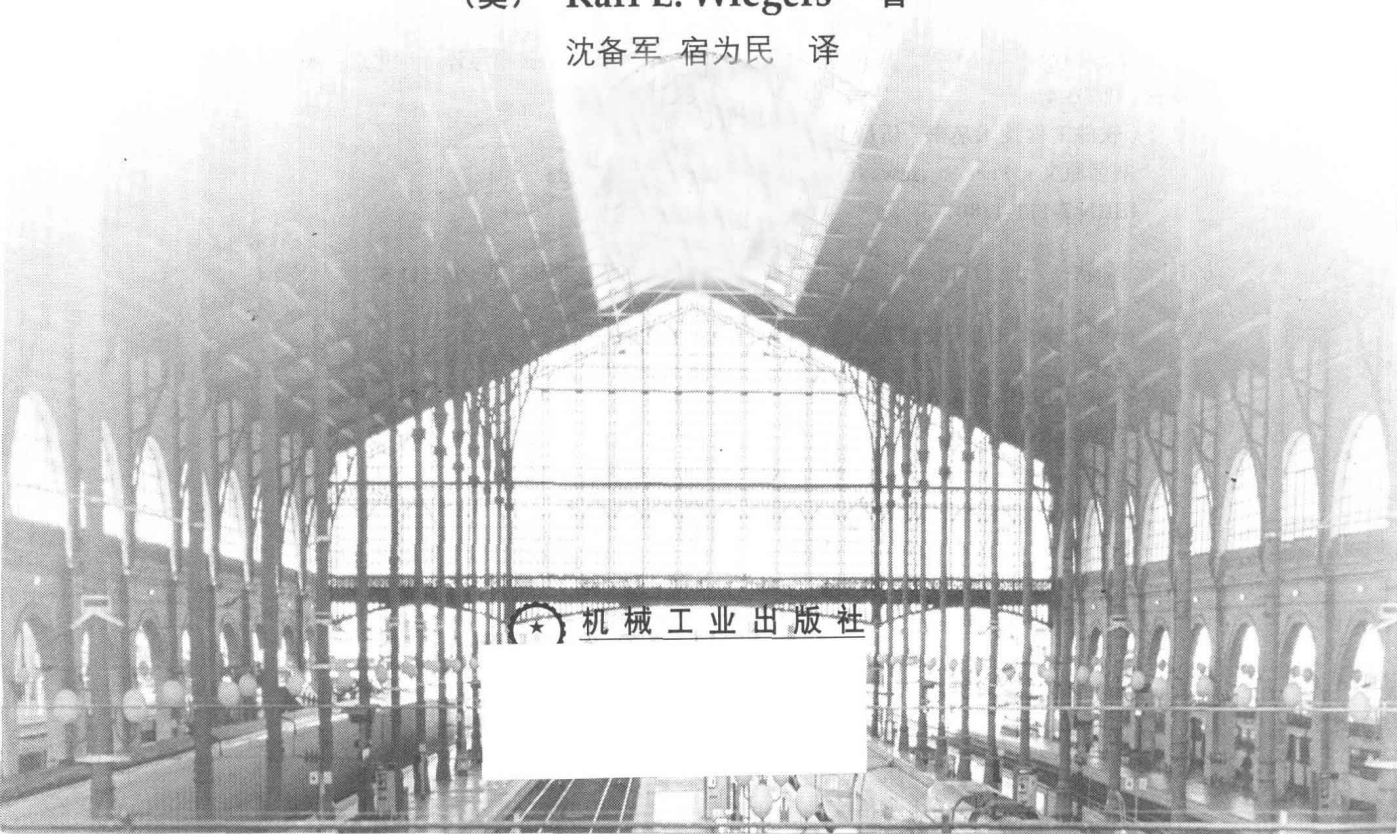
Peer Reviews in Software A Practical Guide

(美) Karl E. Wieggers 著

沈备军 宿为民 译



机械工业出版社



本书论述了软件同级评审的整个过程, 提供了保证软件质量的方法和技术, 涵盖了正式和非正式的评审过程, 描述了多种同级评审方法, 同时对这些方法在各种不同情况下的合理运用提出了建议。另外, 本书还研究了不同的文化和社会差异对评审的影响, 并揭示了建立和实施成功的同级评审计划的关键因素。

本书简明、易懂、实用性强, 适于计算机软件质量管理的从业人员使用。

Authorized translation from the English language edition entitled *Peer Reviews in Software: A Practical Guide* by Karl E. Wiegers, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright © 2002 by Addison-Wesley.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by China Machine Press.

Copyright © 2003 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书版权登记号: 图字: 01-2002-0641

图书在版编目 (CIP) 数据

软件同级评审 / (美) 维杰斯 (Wiegers, K. E.) 著; 沈备军等译. - 北京: 机械工业出版社, 2003.4

(软件工程技术丛书. 质量管理系列)

书名原文: *Peer Reviews in Software: A Practical Guide*

ISBN 7-111-11807-3

I. 软… II. ①维… ②沈… III. 软件质量 - 质量管理 IV. TP311.5

中国版本图书馆CIP数据核字 (2003) 第017230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: 杨文

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2003年6月第1版第1次印刷

787mm × 1092mm 1/16 · 12.5 印张

印数: 0 001 - 5 000 册

定价: 29.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

软件质量和开发进度一直是软件开发成功关键的因素，而在实际工作中只有少量项目能按计划完成，进度要求往往迫使开发组无法保证软件质量，最终许多项目因为质量问题无法投入使用。同级评审，作为一种软件产品验证的活动，能够及早地从软件产品中识别并消除缺陷，从而减少后期的返工，加快开发进度，提高产品质量。这无疑是一种十分有效值得推广的评审方法，也是CMM等级3的关键过程域。

所谓同级评审，就是指由作者的同事对软件开发文档或代码进行评审和检查，帮助查找缺陷和改进的契机。评审者不一定是和作者具有相同开发经验和知识的人员，为了确保质量，应邀请不同背景的评审者从不同的角度检查。管理人员一般不参与同级评审，以免用评审的结果去评估作者绩效。同级评审的目标不仅是捕捉错误、收集改进建议，还可帮助作者在未来创建更好的工程产品。同时，评审还起到了提高项目连续性和培训后备人员的作用。

在组织内成功实施同级评审，不仅需要掌握评审的方法和技术、编写评审计划，而且需要在组织内建立评审的文化。如何避免种种错误并取得成功？本书介绍了软件同级评审的整个过程，描述了正式和非正式的多种同级评审方法，并对这些方法在多种不同情况下的合理运用提出了建议。本书着重描述了审查技术，这是最正式、最严格、最有效的一种同级评审。另外，作者还对评审工作所涉及的文化和社会差异进行了探究，并揭示了在组织中成功建立和实施同级评审过程的关键因素。本书还提供了许多实用的资源，以帮助读者具体实践评审过程。

我们相信，此书的翻译出版将会对促进我国的软件开发和质量管理大有裨益，一定能引起广泛的共鸣。然而，正如周伯生教授所提出的，“软件工程是技术科学、人文科学及实验科学的交叉”。在应用同级评审技术时，应该联系实际，勇于实践，善于总结，结合中国的文化特点和企业的实际情况，切实有效地提高软件质量，改进软件开发过程。

本书第1~7章由沈备军翻译，第8~12章及附录由宿为民翻译。沙科伟、赵丽娟、周德铭、张芳霁协助翻译了部分内容。沈备军和宿为民对全文进行最后统校。本书的校对工作同样采用了同级评审技术，每章进行3~4遍的评审，我们从中也获益匪浅。但因时间仓促且水平有限，疏漏和差错在所难免，敬请读者批评指正。

译者

前言

作为软件开发人员、需求编写者、项目规划者、测试工程师以及书籍作者，这些人无论多么有经验有技能，仍会犯错误。犯错误本身并不错，它是人类生活的一部分。但是一旦错了，就应该尽早地找到错误，不然这些错误会更加难以发现，并且改正起来代价高昂。

对于自己的错误常常因为身在其中反而不易发现。在多年以前，我懂得了让同事检查我的工作并指出其中的错误是非常有价值的。当他们检查时，尽管我总会感到羞怯，但是我更愿意让他们而不是客户来发现错误。这种检查就称为同级评审（peer review）。同级评审可分为多种，如审查、走查等。本书的许多论点是针对以提高质量为目的而进行的产品检查活动，这些活动通常由作者以外的人员来执行。

我于1987年开始实施软件同级评审，今天，如果个工作产品没有经过其他人的仔细检查，我决不会认为这个产品已经完成。人们永远不可能发现全部的错误，但是有了别人的帮助，就可以发现更多的错误。本书的手稿和我以前编写的书都经过了多方面的同级评审，这极大地提高了它们的质量。

本书目标

同级评审没有一种“纯粹的方法”，本书的主要目的旨在帮助有效地评审组织所创建的产品。我也讨论了在软件组织中实施有效的同级评审程序的文化和实践问题。在同级评审中，我特别强调了“审查”这种最正式最有效的方法，但也同时描述了其他几种不同正式程度的方法。所列出的许多参考文献有助于更广泛地了解软件评审和审查。

审查既是软件开发的一个成功故事，又在某种程度上是一种失败。它是成功的，因为它奏效。自从Michael Fagan于1970年在IBM公司提出了审查，它已经成为发现软件错误最有效的方法。读者也不必只听信我的一面之词，不少软件文献已经报道了在许多软件组织中审查是如何提高生产率和产品质量的。但是，却只有一小部分的软件开发团体懂得审查过程，有效实施审查的人就更少。为了有助于读者在小组中实施审查和其他的同级评审，本书将重点放在了任何组织都能应用的实用方法上。

和本书配套的有一个Web网站，从中可以获得一些有价值的过程资源，有助于快速实施同级评审程序。地址如下：

http://www.processimpact.com/pr_goodies.shtml

<http://www.awl.com/cseng/>

这些资源包括评审表格、缺陷检查表、同级评审过程示例、用于收集审查数据的表格和审查的培训等等，参见附录B。欢迎读者下载这些文档，并根据实际需求进行修改。请将你的意见和建议通过kwiegers@acm.org发给我，也欢迎反馈你的同级评审的实施情况。

目标读者

本书提供的材料对许多正在进行项目开发的人是非常有用的，包括：

- 工作产品的作者，包括分析员、设计员、程序员、维护员、测试工程师、项目经理、市场人员、产品经理、技术作家和过程开发者。
- 工作产品的评估者，包括质量工程师、客户代表、客户服务人员和所有作者。
- 过程改进领导。
- 上述这些人员的管理者，他们需要知道如何将同级评审引入到他们的文化中，并且让所发布的产品得到评审。

本书将帮助那些已认识到其软件产品缺少质量目标，以及那些想提高他们当前评审实践的人，确保他们在项目中建立和维持良好的沟通，或按时交付高质量的软件。正在软件方面采用能力成熟度模型或在系统工程/软件工程方面采用CMMI的组织将会发现本书很有价值，因为同级评审是这些过程改进框架的组成部分（参见附录A）。

本书描述的技术不限于软件项目的可交付产品和文档。事实上，任何工程项目的技术工作产品都可以应用，包括设计规格说明、图表、装配指令和用户手册等。任何一个业务，如果具有文档编制任务过程或质量控制过程，都会发现认真的同级评审能找到作者自己无法轻易发现的错误。

阅读建议

若要更详细地了解一般的同级评审方法和特殊的审查，读者可以简单地把本书从头读到尾。同级评审的文化和社会问题在第1章和第2章进行了讨论；第3章概述了不同类型的评审方法，以及对如何选择合适的评审方法提供了建议；第4~8章详细讨论了审查过程；第9章描述了重要的审查数据项和度量。如果读者想在组织中实施成功的评审程序，请集中研究第10章和第11章。如何面对特殊评审的挑战，如大件工作产品或分布式开发小组，请参考第12章中的建议。本书用到的许多术语的定义请参见术语表。

关于作者

Karl E. Wiegers是软件工程领域公认的权威。他是Process Impact 公司的首席顾问，这是一家软件过程咨询和培训公司，位于俄勒冈州的波特兰。他的兴趣包括软件质量工程、需求工程、项目管理、风险管理、度量和软件过程改进。在此之前，他在柯达公司工作了8年，担任图像研究科学家、软件开发人员、软件经理、软件过程和质量改进的领导。Karl 获得Boise 州立学院的化学专业的学士学位，接着在伊利诺依大学获得了有机化学的硕士和博士学位。他是IEEE、IEEE计算机协会和ACM的成员。

Karl是《软件需求》(中文版已由机械工业出版社出版)和《创立软件工程文化》的作者，这两本书都荣获了《Software Development》杂志的效率大奖。另外他还撰写了140多篇关于计算、化学和军事历史的文章。他还是《IEEE Software》杂志的编委会成员和美国《Software Development》杂志的特约编辑。他经常在软件会议和专业社团会议上演讲。工作之余，Karl喜欢烹饪和葡萄酒，研究军事历史，并喜爱骑铃木Vx800摩托和弹吉它。

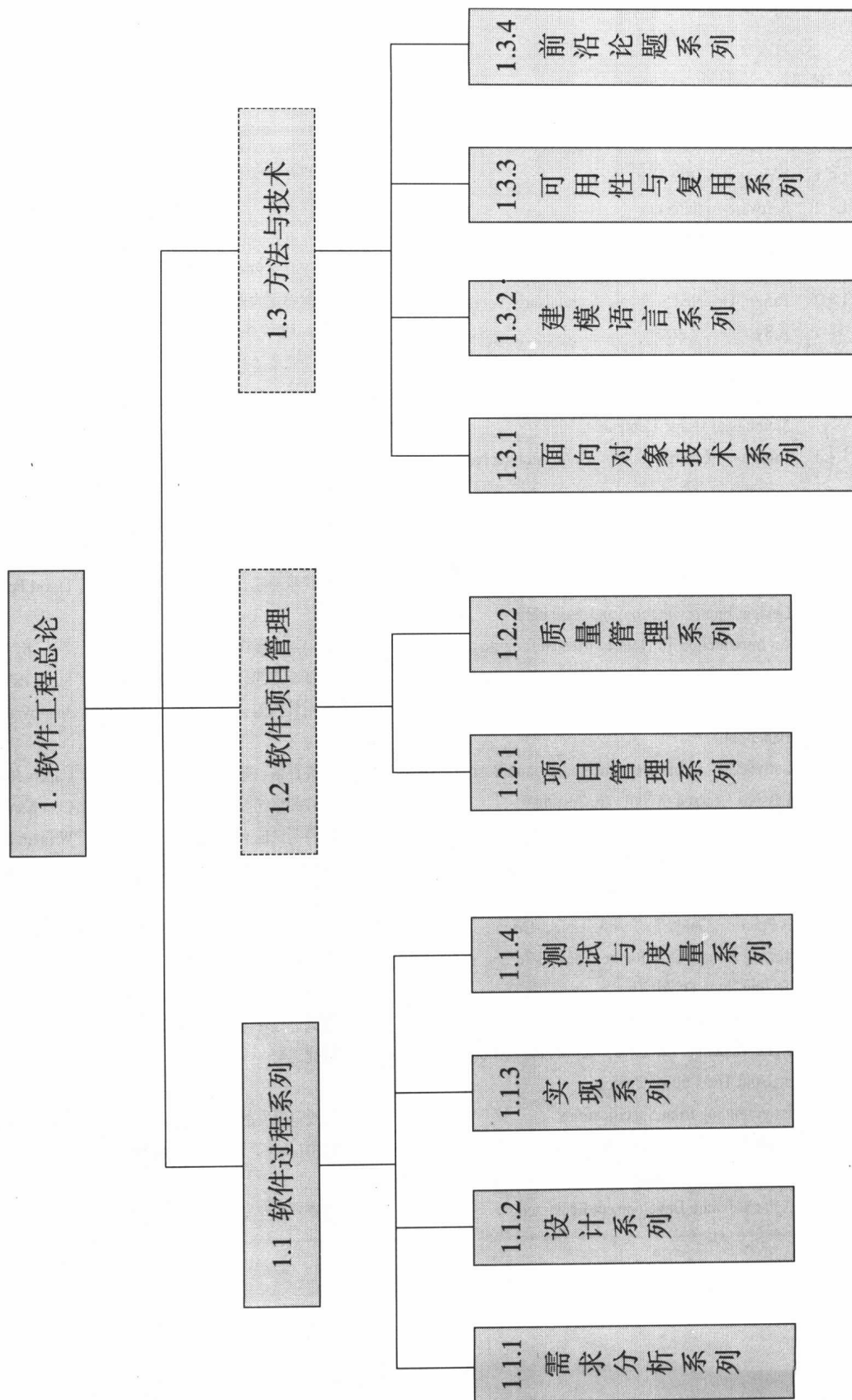
软件工程丛书书目

丛书 编号	英文书名	中文书名	作 者
1	"Software Engineering A Practitioner's Approach, 5E	软件工程：实践者的研究方法 (原书第5版)	Roger S. Pressman
1	Software Engineering	软件工程	Ian Sommerville
1	Software Engineering with Java	软件工程——Java语言实现	Stephen R. Schach
1	Object-Oriented Software Engineering	面向对象的软件工程	Timothy C. Lethbridge
1.1.1	Software Process Improvement	软件过程改进	Sami Zahran
1.1.1	Successful Software Development	成功的软件开发	Scott Donaldson
1.1.1	Making Process Improvement Work	成功的过程改善	Neil S. Potter
1.1.1	Implementing the Capability Maturity Model	CMM实施指南	James R. Persse
1.1.1	CMMI Distilled:A Practical Introduction to Integrated Process Improvement	CMMI 精粹——集成化过程 改进实用导论	Dennis M. Ahern
1.1.1	CMM Implementation Guide	CMM实施指南	Kim Caputo
1.1.2	The Road to the Unified Software Development Process	统一软件开发过程之路	Ivar Jacobson
1.1.2	The Unified Software Development Process	统一软件开发过程	Jacobson/Rumbaugh
1.1.2	UML and The Unified Process Practical Object-Oriented Analysis & Design	UML和统一过程：实用的面向 对象的分析和设计	Jim Arlow
1.1.2	The Rational Unified Process: An Introduction Second Edition	Rational统一过程引论 (原书第2版)	Philippe Kruchten
1.1.3	Software Project Management: A Unified Framework	软件项目管理——一个统一的 框架	Walker Royce
1.1.3	AntiPatterns and Patterns in Software Configuration Management	软件配置管理中的模式与反模式	William J. Brown
1.1.3	Managing Global Software Projects	管理全球软件项目	Remesh
1.1.3	Software Project Management	软件项目管理	Bob Hughes
1.1.3	IT Project Management	IT项目管理：从开始到结束的 历程	Joseph Phillips
1.1.3	How to Run Successful Projects III The Silver Bullet	成功的项目管理	Fergus O'Connell
1.1.3	Architecture-Centric Software Project Management	以体系结构为中心的软件项目 管理	Daniel J. Paulish
1.1.4	Peer Reviews in Software: A Practical Guide	软件同级评审——实践指南	Karl E. Wiegiers
1.1.4	Handbook of Software Quality Assurance	软件质量保证	Gordon Schulmeyer
1.1.4	Performance Solutions A Practical Guide To Creating Responsive, Scalable Software	软件性能方案	Connie U. Smith
1.1.4	Software Reliability Engineering	软件可靠性工程	Musa

丛书 编号	英文书名	中文书名	作 者
1.1.4	An Introduction To Reliability And Quality Engineering	可靠性与质量工程	John Bentley
1.1.4	Objece-Oriented Defect Management of Software	面向对象软件的缺陷管理	Houman Younessi
1.1.4	Metrics and Models in Software Quality Engineering	软件质量工程的度量和模型	Stephen H. Kan
1.1.4	Customer Oriented Software Quality Assurance	面向用户的软件质量保证	Frank P. Ginac
1.1.5	Software Metrics: A Rigorous and Practical Approach	软件度量 (原书第2版)	Norman E. Fenton
1.2.1	The Object Primer: The Application Developer's Guide to Object Orientation and the UML	面向对象和UML应用开发指南	Scott W. Ambler
1.2.1	Object-Oriented Methods Principles & Practices	面向对象的方法: 原理与实践 (原书第3版)	Ian Graham
1.2.1	Project-Based Software Engineering: An Object-Oriented Approach	基于项目的软件工程——面向对象研究方法	Evelyn Stiller
1.2.1	The Unified Modeling Language User Guide	UML用户指南	Booch/Rumbaugh/Jacobson
1.2.1	Applying UML and patterns:An Introduction to Object-Oriented Analysis and Design	UML和模式应用	Craig Larman
1.2.1	The Unified Modeling Language Reference Manual	UML参考手册	Rumbaugh/Jacobson/Booch
1.2.1	Object Solutions Managing the Object-Oriented Project	面向对象项目的解决方案	Grady Booch
1.2.1	Business Modeling with UML: Business Patterns at Work	用UML进行商业建模	Hans-Erik Eriksson
1.2.2	Practical Software Reuse	实用软件复用方法	Donald J. Reifer
1.2.2	Software Reuse Techniques: Adding Reuse to the Systems Development Process	软件复用技术: 在系统开发过程中考虑复用	Carma McClure
1.2.2	Software Reuse: Architecture, Process and Organization for Business Success	软件复用: 结构、过程和组成	Jacobson/Griss
1.2.2	Component-based Product Line Engineering with UML	基于构件的产品线工程: UML方法	Colin Atkinson
1.2.2	Large-Scale, Component-Based Development	大规模基于构件的软件开发	Alan W. Brown
1.4.1	Effective Requirements Practices	有效需求实践	Ralph R. Young
1.4.1	Applying Use Cases, 2E	用例分析技术 (原书第2版)	Schneider
1.4.1	Requiremements Engineering: A Good Practice Guide	需求工程	Ian Sommerville
1.4.1	Requirements Analysis and System Design: Developing Information Systems with UML	需求分析与系统设计	Leeszek A. Maciaszek
1.4.1	Systems Analysis and Design Methods	系统分析与设计方法	Whitten
1.4.1	Systems Analysis and Design in a Changing World	系统分析与设计	Satzinger/Jackson/Burd
1.4.1	Software Requirements and Estimation	软件需求与估算	Kishore
1.4.1	Managing Software Requirements	软件需求管理: 统一方法	Dean Leffingwell
1.4.1	Object-Oriented Analysis and Design with Applications	面向对象的分析与设计——实例应用	Grady Booch
1.4.1	Advanced Use Case Modeling Software Systems	高级用例建模, 卷1: 软件系统	Frank Armour
1.4.1	Analysis Patterns Reusable Object Models	分析模式: 可复用的对象模型	Martin Fowler

丛书 编号	英文书名	中文书名	作 者
1.4.1	Writing Effective Use Cases	编写有效用例	Alistair Cockburn
1.4.2	Software Architecture	软件架构：组织原则与模式	David Dikel
1.4.2	Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects	面向模式的软件体系结构 卷2：用于并发和网络化对象的模式	Douglas Schmidt
1.4.2	Pattern-Oriented Software Architecture, Volume 1: A System of Patterns	面向模式的软件体系结构 卷1：模式系统	Frank Buschmann
1.4.2	Object-Oriented Software Construction	面向对象的软件结构	Bertrand Meyer
1.4.2	Software for Use: A Practical Guide to The Models and Methods of Usage-Centered Design	Software for use (中文版)	Larry L. Constantine
1.4.2	Patterns of Enterprise Application Architecture	企业级应用程序体系结构中的模式	Martin Fowler
1.4.3	Principles of Object-Oriented Software Development	面向对象软件开发原理 (原书第2版)	Eliens
1.4.3	Model-Driven Development: Automating Component Design, Implementation, and Assembly	模式驱动的开发	David Frankel
1.4.3	An Introduction To Object-Oriented Programming	面向对象程序设计引论	Timothy Budd
1.4.3	A Practical Guide to Feature-Driven Development	特征驱动的软件开发指南	Steve Palmer
1.4.4	Testing IT: An Off-the-Shelf Software Testing Process Handbook	实用软件测试过程之路	John Watkins
1.4.4	Software Assessments, Benchmarks, and Best Practices	软件评估、基准和实践	Capers Jones
1.4.4	Lessons Learned in Software Testing	软件测试经验与教训	Cem Kaner
1.4.4	Effective Methods for Software Testing	软件测试的有效方法	William E.Perry
1.4.4	Software Testing in the Real World: Improving the Process	软件测试：实战中的过程改进	Edward Kit
1.4.4	Beta Testing for Better Software	软件Beta测试	Michael R. Fine
1.4.4	A Practical Guide To Testing Object-Oriented Software	面向对象的软件测试	John D. McGregor
1.4.4	Testing Computer Software: The Bestselling Software Testing Book Of All Time!	计算机软件测试	Cem Kaner
1.4.4	Managing the Testing Process, 2E	测试过程管理 (原书第2版)	Rex Black
1.5	Agile Modeling: Effective Practices For eXtreme Programming And The Unified Process	敏捷建模	Scott Ambler
1.5	Programming from Specifications	从规范出发的程序设计	Carroll Morgan
1.5	JAVA Tools For Extreme Programming: Mastering Open Source Tools Including Ant, Junit, and Cactus	应用JAVA工具进行极限编程	Richard Hightower
1.5	Agile Software Development Ecosystems	敏捷软件开发的生态系统	Tom DeMarco

软件工程技术丛书结构图



目 录

译者序

前言

关于作者

第1章 质量的挑战1

- 1.1 查看他人的工作1
- 1.2 质量并不免费3
- 1.3 合理评价同级评审5
- 1.4 同级评审、测试和质量工具7
- 1.5 能够评审什么9
- 1.6 对质量的个人承诺10

第2章 来自朋友的帮助13

- 2.1 查找别人的错误13
- 2.2 评审和小组文化14
 - 2.2.1 文化的影响16
 - 2.2.2 评审与管理者17
 - 2.2.3 为什么人们不愿意实施评审20
 - 2.2.4 克服对评审的抵触情绪21
- 2.3 同级评审的评审级别24
- 2.4 为评审制定计划26
- 2.5 评审的指导原则27

第3章 同级评审方法的正式化频谱29

- 3.1 正式化频谱29
 - 3.1.1 审查31
 - 3.1.2 小组评审32
 - 3.1.3 走查33
 - 3.1.4 结对编程35
 - 3.1.5 同级桌查36
 - 3.1.6 轮查37
 - 3.1.7 临时评审38
- 3.2 选择合适的评审方法38

第4章 审查过程41

- 4.1 审查角色41
 - 4.1.1 作者角色42
 - 4.1.2 读或不读43
- 4.2 审查小组的规模44
- 4.3 审查过程的各个阶段45
 - 4.3.1 制定计划47
 - 4.3.2 总体会议47
 - 4.3.3 准备48
 - 4.3.4 会议48
 - 4.3.5 返工50
 - 4.3.6 跟踪50
 - 4.3.7 因果分析51
- 4.4 不同的审查方案51
 - 4.4.1 Gilb/Graham方法51
 - 4.4.2 High-Impact审查53
 - 4.4.3 分阶段审查53

第5章 制定审查计划55

- 5.1 何时进行审查56
- 5.2 审查的评审组长58
- 5.3 选择审查材料60
- 5.4 审查准入条件61
- 5.5 汇集各方观点62
 - 5.5.1 审查者的视角63
 - 5.5.2 管理者和观察者66
- 5.6 审查包67
- 5.7 审查速率68
- 5.8 制定审查活动进程表70

第6章 检查工作产品73

- 6.1 总体阶段73
- 6.2 准备阶段75
- 6.3 准备的方法78

6.3.1 缺陷检查表	79	9.6.3 投资回报	124
6.3.2 规则集	80	第10章 建立同级评审程序	127
6.3.3 其他分析技术	81	10.1 同级评审过程拥有者	127
第7章 齐心协力进行审查	85	10.2 组织准备	128
7.1 评审组长的角色	85	10.3 过程资产	132
7.2 启动审查会议	89	10.4 同级评审协调者	134
7.3 举行会议	91	10.5 同级评审培训	135
7.3.1 读工作产品	91	10.6 实验评审过程	139
7.3.2 发现缺陷和问题	93	第11章 让同级评审发挥作用	141
7.3.3 记录缺陷和问题	95	11.1 关键成功因素	141
7.3.4 观察问题	97	11.2 需避免的评审陷阱	144
7.4 产品决议	101	11.3 评审问题的解决	146
7.5 结束会议	102	第12章 特殊评审的挑战	151
7.6 改进审查过程	103	12.1 大件工作产品	151
第8章 结束审查	105	12.2 空间和时间上的分离	152
8.1 返工阶段	105	12.2.1 分布式评审会议	154
8.2 跟踪阶段	107	12.2.2 异步评审	155
8.3 因果分析阶段	109	12.3 生成的和非过程的代码	156
8.4 审查准出条件	110	12.4 有太多的参与者	157
第9章 分析审查数据	113	12.5 缺乏合格的评审人员	158
9.1 为何要收集数据	113	尾声	159
9.2 关于测量的说明	114	附录A 软件同级评审及过程改进模型	161
9.3 基本数据项和度量	116	附录B 补充资料	171
9.4 审查数据库	117	同级评审的术语表	173
9.5 数据分析	119	参考文献	179
9.6 测量审查的效果	122		
9.6.1 有效性	122		
9.6.2 效率	124		

第 1 章

质量的挑战

“嗨，Maria，你有时间吗？我找不到程序中的一个小缺陷，你能帮我看一看这段代码吗？”

“当然可以，Phil，哪里有问题？”

“这些图像没有正确对齐，它们应该全部左对齐，但每一个都是缩进排列。我确信问题出在 DisplayImage 函数，但我已研究了15分钟，没能发现任何错误。”

“嗯，让我看看。它好像……” [细细低语] “不，那部分是对的，但请看这个循环的起始处。” Maria指着屏幕说，“我认为这个括号放错了位置。如果你将索引变量移至外面，图像将不会缩进排列。”

Phil用手拍着额头，说道“你是对的！真谢谢你，Maria，我简直不能相信我没有发现那个错误。”

“太客气了，Phil，我很乐于帮忙。”

许多程序员都曾经请求同事帮助寻找代码中不易发现的问题。因为身在其中而常常不能发现自己的错误。当研究代码时，你的大脑正在重复原有的思维活动，以同样的原因重复原有的错误。你需要从一个新的角度出发：一双从未看过这些代码的眼睛和一个以不同思维方式思考的大脑。

1.1 查看他人的工作

在同级评审中，由生产产品的作者以外的其他人来检查工作产品，发现缺陷，并寻找改进的契机。所谓缺陷（defect，或错误），是指软件工作产品中的一种情况，它将导致软件产生不令人满意或非预期的后果。当Phil陷入困境时，他请Maria来做一次简单的代码评审。Maria只用几分钟的时间研究了问题代码，然后就将他从困境中解救出来。即使你的组织中采用了正式的同级评审过程，还是要依靠同事的热心帮助来进行这些临时的快速的评审。

为了发现缺陷，人们采用了包括同级评审在内的多种项目评审，表1-1给出了其中的一些方式。虽然所有的评审对项目的成功都有帮助，并且也可能涉及作者的“同级同事”，但本书的重点在于同级评审，其首要目的就在于提高产品的质量。

表1-1 项目评审的类型

评审类型	目 的
教育评审	让其他项目相关人员来督促与项目相关的技术论题
管理、预备、或准入评审	向上层管理者提供信息，以帮助他们做出如下决策：发布产品、继续（或取消）开发项目、批准（或拒绝）提案、改变项目范围、调整资源、改变承诺
同级评审	寻找工作产品中的缺陷和改进契机
项目后的评审	对最近完成的项目或阶段进行评审，让未来的项目吸取经验
状态评审	向项目经理和其他项目成员提供最新的项目状态信息，包括里程碑的进度、所遇的问题、识别的或受控的风险

术语“同级评审”有时被误认为是对同事的绩效评估。事实并非如此，任何同级评审都是针对工作产品，而不是创建工作产品的个人。同级评审已成为科学和工程工作的基本构件。一个科学家的同事能够判断工作是否满足专业标准，能够在实验或设备设计、数据收集或结果分析中发现错误。科学的结论只有当其通过了同级评审之后才会被考虑其有效性。这对于软件开发也是一个不坏的哲理。多种广泛使用的过程改进框架都把同级评审作为组织必须采用的关键实践，以提高软件工程能力，详见附件A。

当我使用“评审”或“同级评审”术语时，是指软件同级评审的总体活动，而不精确地考虑它如何具体进行。术语“评审”、“审查”和“走查”有时可交换使用，但它们代表了同级评审的不同方法。第3章叙述了从非正式评审（如Phil和Maria的经历）到高度系统化和规范化的审查过程的一系列同级评审方法。第3章也建议了专门的评审技术，以满足特殊项目的要求。

评审的目标不仅是捕捉错误，收集改进的建议，而且帮助作者在未来创建更好的工作产品。我曾经以作者或评审者的角色参与了多次评审，从中学到了不少。同级评审的好处是如此突出，以致我不愿在不设同级评审的组织中工作。

1.2 质量并不免费

管理者、开发人员和客户有时反对同级评审，因为他们认为评审的花费太大，并将减慢项目的进度。事实上，评审并不会减慢进度，缺陷却会。任何一个经历过疯狂的测试-调试过程的人都会认识到后期发现缺陷的昂贵代价，这将使得最终产品延期交付。只有当你的工作产品在评审时未能发现缺陷，评审才是浪费时间。

可能你听到过质量是免费的论点（Crosby 1979）。这有些过于简单化了。“质量是免费的”意味着你在产品质量上的投资超过了减少后期缺陷和客户报告的问题的回报。另一种观点是将缺陷对项目和产品的影响考虑为质量成本，它包括以下几类活动所花的时间和资金：

- 外部产品失效，包括处理客户报告的缺陷，开发代码补丁或未列入计划的发布，以及实现遗漏的功能。
- 内部产品失效，例如纠正在测试或产品发布前同级评审发现的缺陷，缺陷管理，缺陷构件的返工，和重新测试修改后的构件。
- 质量评估，如执行评审或测试来发现缺陷，收集和分析质量度量。
- 缺陷预防，包括培训开发人员，分析缺陷产生的原因，开发和改进工程和质量过程。

返工——重做你认为已经完成的事情所花的时间——是一个项目质量成本的最主要部分。如果许多小的纠正活动花费了数小时的工作，返工就会对软件开发的生产率造成很大损害。根据一些企业返工度量的报道，它可占整个开发工作量的40%~60%（C. Jones 1986; Cooper and Mullen 1993; Haley 1996; Wheeler, Brykczynski, and Meeson Jr. 1996b）。虽然评审消耗了资源，不是免费的，但在评审上所花的时间能够减少大量项目后期的返工。那些被避免的返工时间可用于开发工作，这对你的顾客和业务都是十分值得的。

在同级评审上的投入把组织的一些质量成本从昂贵的、后期的返工转变为早期的缺陷发现。更重要的是，工作产品的作者学到了如何将工作做得更棒，从而避免了缺陷。不管你有没有发现它们，缺陷总是存在的，问题只是当最终发现它们时，需要多少纠正成本。例如，

Raytheon电子系统在两年中将返工率从项目总开支的41%降低到20%，这在很大程度上归功于执行了审查程序（Haley 1996）。Raytheon 同时还减少了80%代码集成过程中问题纠正的工作量，并削减了一半的重新测试工作量。评审同样也改进了开发过程，通过缺陷预防提高了组织的效率，这是一种巨大的战略利益。预防缺陷比你完成实现或交付产品后再去除它们更便宜。

在开发后期或在产品发布后发现并纠正缺陷的费用很高，因而在产品开发早期检测缺陷将有很大的潜在回报。航天飞机搭乘软件项目的测量结果是，如果在审查时纠正设计或代码的相对成本为1美元，那么在系统测试时为13美元，在交付后则为92美元（Paulk et al. 1995）。另一些研究表明，纠正客户报告的与需求相关的缺陷的费用是在需求开发阶段发现和纠正同样缺陷的68倍~110倍（Boehm 1981;Grady 1999）。

我的某个咨询客户是一个电信公司，它们利用审查发现和纠正一个缺陷的平均费用为200美元，而改正一个客户发现的缺陷平均花费4200美元。这个放大倍数对于不重要的小系统没有这么严重，但绝对不会是零（Boehm and Basili 2001）。你可以用评审来评价产品质量的其他方面，如可靠性、健壮性、可维护性和可测试性。如果创建和维持评审程序、培训团队和执行评审的费用（投入）低于由于减少返工而节约的费用和由于提高客户满意度所产生的回报，那么你就已经迈出了一大步。

每个项目组必须在初期产品质量和其他业务目标的投入之间取得平衡，其他业务目标包括上市时间、产品特性和长期的维护费用等。大家都认为，对于一个以进度驱动的项目而言，评审是它所不能负担的奢侈行为。是的，评审需要花费时间，但是，是否它们花费得太多，则是商务问题，当评审应用得好时，它实际上可以通过绕过一些测试阶段来缩短产品开发进度（Weller 1993）。

从长远利益看，关于质量的实践活动可以更快地推出“足够好”的产品。有时管理人员或市场人员却决定不执行质量活动，他们在试图满足紧迫的交付进度与付出更多的资源进行问题纠正之间进行了商务折衷。这样的决策必须基于对质量成本及其蕴含的商务可能性所进行的风险评估。如果管理者认识到长期的质量需求，他们就会在项目进度中安排评审的时间。