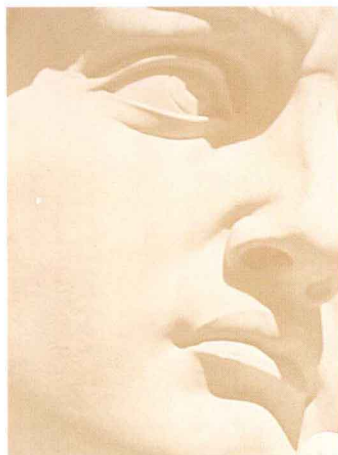




软件工程的本质

运用SEMAT内核 Ivar Jacobson Pan-Wei Ng Paul E. McMahon
Ian Spence Svante Lidman 著 陈钟 等译



APPLYING THE SEMAT KERNEL

软件工程引擎返璞归真，回归软件工程基本层面
全球软件社区广泛支持，全新标准已获OMG通过
SEMAT发展的重要里程碑，软件工程领域的大事件



机械工业出版社
China Machine Press

软件工程的本质

运用SEMAT内核

Ivar Jacobson Pan-Wei Ng Paul E. McMahon
Ian Spence Svante Lidman 著 陈钟 等译

APPLYI

RNEL



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

软件工程的本质: 运用 SEMAT 内核 / 雅各布森 (Jacobson, I.) 等著; 陈钟等译. —北京: 机械工业出版社, 2013. 8

书名原文: The Essence of Software Engineering: Applying the SEMAT Kernel

ISBN 978-7-111-43280-7

I. 软… II. ①雅… ②陈… III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2013) 第 154893 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2013-1408

Authorized translation from the English language edition entitled The Essence of Software Engineering: Applying the SEMAT Kernel, First Edition (ISBN-13: 978-0-321-88595-1) by Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence, Svante Lidman, Copyright © 2013 by Ivar Jacobson, Pan-Wei Ng, Paul McMahon, Ian Spence, Svante Lidman.

Chinese simplified language edition published by China Machine Press.

Copyright © 2013 by China Machine Press.

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 the prior written permission of the publisher.

SEMAT (软件工程方法学) 是一个国际倡议, 旨在确定一个软件工程的公共基础或通用标准。通过创建一种简单的语言来描述方法和实践, SEMAT 团队将这个公共基础表示为一个由所有软件开发基本要素组成的内核 (或框架)。

本书由 SEMAT 的创始人和思想贡献者编写, 阐述了软件工程的本质, 介绍了 SEMAT 内核, 分享了运用 SEMAT 内核解决具体问题的案例和有效性。它是 SEMAT 新标准的纲领性著作, 也是一本引导我们重新认识软件、软件工程和人的关系的书, 适合软件从业人员、软件工程学者、教师和学生阅读、参考和学习。

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: 朱秀英 姚蕾

北京市荣盛彩色印刷有限公司印刷

2014 年 1 月第 1 版第 1 次印刷

170mm × 242mm · 16 印张

标准书号: ISBN 978-7-111-43280-7

定 价: 55.00 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) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

本书终于和大家见面了，作为 SEMAT 中国的主席和本书翻译组的组长，我感到非常高兴。这是 SEMAT 发展的一个里程碑，也是软件工程领域的一件大事。

本书英文版在写作过程中就受到 SEMAT 社区的广泛关注。英文版定稿不久 SEMAT 国际就积极地组织其他各国语言版本的翻译，中文版是最先启动的，其他还包括西班牙文版、俄文版等。我们也和大家一样，怀着兴奋、好奇和忐忑不安的心情开始阅读原著，并力求在深入理解 SEMAT 精髓理念的基础上同时把原英文版的文字准确、完整地映射到我们熟悉的汉语中，追求“信达雅”的翻译目标。然而，由于东西方文化的差异，直至完成译稿之后还是感觉有许多不尽如人意的地方。

首先，我们很愿意向软件开发者和软件工程学者推荐这本书。这是一本与众不同的书，是引导我们重新认识软件、软件工程和人的关系的书。软件工程从 1960 年 NATO 会议上首次提出，已经走过 50 多年的历程，也出版过许多

相关的书籍。但多数高校使用的软件工程教材强调软件产品开发与维护的工程属性和训练，而现实工作中软件开发者遵循在学校很少开课的敏捷开发方法。业界号称各种软件开发的模型和方法多如牛毛，甚至高级软件开发主管也感到无所适从。问题究竟出在哪里？正像我的导师杨芙清院士所指出的：软件是人们智慧的产品，兼具工程和艺术属性，因此软件工程师需要有数学的思维、工程的严谨和艺术的创造三位一体的训练。杨院士这一高屋建瓴的见解也成为北京大学创办软件学院，培养软件工程复合型、创新性人才的指导思想。我作为北京大学软件学院的创始院长，从 2002 年起在历时八年的任职工作中，跟随杨院士努力创建一个工程和艺术交融的软件工程教育环境，促进软件工程人才的成长，办学实践不仅得到国家级教学成果一等奖，更重要的是得到了社会的认可。我从北京大学软件学院院长岗位卸任后，恰逢 Ivar 先生发起 SEMAT 倡议后来中国与中国软件工程届交流并获得杨芙清院士的认可，在杨院士的推荐下我参与了早期的准备工作并担任了 SEMAT 中国主席，与国内软件工程同行共同反思软件工程理论与方法学。通过参与 SEMAT 国际化的研究与讨论，特别是和国际大师们的接触和交流，结合自己的工程实践，我看到工匠文化深厚的欧洲人对软件的工程与艺术不仅有深刻的认识，而且从理论的层面把工程和艺术有机结合起来，让理论、方法和实践有机统一，让开发、维护和使用软件的人们在一个共同的理论基石上协同发展。这或许就是 SEMAT 的与众不同之处。它没有教给你一个具体的新方法，但却教给你一个重新思考并有效建立软件与人的新体系的工具。更为可喜的是，SEMAT 国际同仁们并不止步在 SEMAT 思想，而是经过不懈努力让 SEMAT 内核成为 OMG 的标准，进一步规范和指导软件工程实践，并为推动人们进一步丰富和发展 SEMAT 奠定了理论框架和技术基础。在本书出版之际，我们已经高兴地看到关于 SEMAT 内核标准的最后一轮 OMG 投票已获得通过。

这本书阐述了 SEMAT 的本质，分享了运用 SEMAT 内核解决具体问题的案

例和有效性,虽然篇幅有限涵盖不了更多的内容,但它是 SEMAT 的一个里程碑。我相信这本书对众多软件从业人员和软件工程学者都有重要的参考价值。

其次,关于本书术语的翻译和理解。东西方文化的差异导致术语的选取和理解可能会存在差异或者空白,这个问题很难仅仅从字面翻译上来对应。举例来说,SEMAT 在设计内核时选择了希腊字母阿尔法作为一个重要的构件,对于一个中国人来说,当你第一次听到或看到阿尔法时,你会想到什么?对于欧洲人可以很自然从字母表的位置常识理解它是第一和重要的概念,但对于中国人它更像一个抽象的概念而没有一个常识性的背景。事实上,阿尔法在 SEMAT 标准中也是经过反复斟酌最终确定的一个术语,之前软件工程界并没有使用过这个术语。在本书中我们选择直译为“阿尔法”,虽然感到不甚满意,但也只能如此,希望读者深入了解全部内容之后能够形成相应概念。

本书的另一个特点是建立关于内核阿尔法及其状态的概念群,其中有许多概念具有敏捷文化色彩。众所周知,大量敏捷开发方法及其术语的中文翻译是非常不统一的,以至于许多中文文档干脆选择直接使用英文单词或短语表达而不去翻译,本书中提及的 Scrum、spike 等。

书中介绍内核的七个阿尔法:包括 Opportunity、Stakeholder、Requirements、Software System、Work、Team、Way of Working,大部分我们可以形成共识,其中 Opportunity 翻译成“机会”而没有称为“商机”,泛指所有商业或非商业目标的开发;Stakeholder 意指“利益相关者”、“干系人”,但为了出现在卡片上更为简明,本书采用了“涉众”的译法。

第三,关于本书的翻译组织。本书采取了一种基于迭代方法的新型翻译模式,有 8 位成员有幸参与到翻译过程中,阅读原著,之后平行翻译各个章节的第一版,在第一版的基础上通过研究生讨论班讲解各自的理解,对概念、方法、实例等进行深入理解,之后形成术语表并迭代修订出第二版。然后重新分工进行交叉阅读,修订形成第三版。在第三版的基础上重新进行翻

译和精化，集中统一校对，对归纳出的共性问题进行讨论，并通过举行中国 SEMAT 研讨会和学术界及工业界同仁们进行交流和研讨，最终定稿。

在这里，我要感谢翻译组中的每一位成员（马擎宇、梁卫国、沈玮、袁子晴、蔡飞志、周濛、汪博、郑聃威）参与初稿翻译，研究讨论班的讲解和实践，以及交叉阅读所做出的贡献。感谢马擎宇在翻译过程中为协助整理术语表和集中统一格式排版所做出的贡献。

第四，致谢。我要代表翻译组感谢复旦大学赵文耘教授，北京航空航天大学刘超教授和吴玉美副教授，清华大学刘强副教授和白晓颖教授，武汉大学何璐璐博士，南京大学陶先平教授，上海大学缪淮扣教授，北京大学孙艳春教授等，他们不仅代表国内 7 所大学支持直接参与了 SEMAT 的研究和教学的活动，而且通过阅读原著、参加 SEMAT 研讨会等为本书的翻译和定稿给予了大力的指导。

我要代表翻译组感谢 IJI 公司的团队参加 SEMAT 中国研讨会并提供相关资料对翻译中文稿提出宝贵的意见。我还要感谢山东潍坊高新技术产业开发区管委会张龙江副主任专程来北京参加 SEMAT 中国研讨会并在近三年来对 SEMAT 发展的关注与支持。感谢中创软件陈志平副总裁、软件行业协会系统与软件过程改进分会王钧秘书长、CSDN 蒋涛总裁等对 SEMAT 中国的积极推进支持。我要感谢美国佛罗里达州大西洋大学的 Shihong Huang 教授和原书作者之一 Pan-Wei Ng（黄邦伟）先生对 SEMAT 中文术语和概念的有益讨论。

我还要特别感谢 Ivar 先生在推进 SEMAT 标准化以及本书翻译过程中通过 Skype 所进行的跨时空的、经常的、细致的讨论，不论他在欧洲、美国、中国或开曼群岛的任何时区，都能够让我和翻译组的同学们近距离感受到这位大师的真知灼见和工作热情，也为更好地拟合东西方文化差异提供了非常有益的帮助。

感谢机械工业出版社对本书出版所给予的大力支持。

最后，进一步与读者交流与互动。本书的翻译只是 SEMAT 中国工作的一个开端，我们还将进一步组织 OMG 标准及其他重要相关文献的翻译出版，以期为中国广大的软件开发者提供最好的帮助。欢迎广大读者通过 SEMAT 中文网站 www.semat.org.cn 或英文网站 www.semat.org 关注 SEMAT 的最新进展，积极反馈意见并分享各种工具和资源。

陈钟

2013 年 4 月

Robert Martin

作序

THE ESSENCE OF SOFTWARE ENGINEERING
Applying the SEMAT Kernel

钟摆再次摆动。这次它摆向了软件工艺，作为软件工艺运动的领导人之一，我想这是一件好事。我认为很重要的一点是软件开发人员应该向其他手工艺者学习“为自己拥有的技艺感到自豪”的精神。

但是当钟摆摆向一边时，也常常摆离了另一边。此次好像摆离了工程概念。这一观点像是在说：如果软件是一件手工艺品或艺术作品，那么它就不能是一门科学或一门工程学科了。我非常不赞成这种观点。

软件既是艺术又是科学产物，它同时需要激情和规范。编写好的软件既需要想象力和创造力的自由翱翔，又需要残酷现实的工程权衡。软件，就像其他值得人们付出努力的事情一样，是左、右大脑之间的混合。

本书尝试描述这种平衡。它提出了一个符合工程规范要求的软件工程框架或“内核”（kernel），同时为工艺所需要的创造性和突发灵感留下了发展空间。

大多数关于软件过程的描述都将其比喻为一条装配生产线。项目沿着装配线从一个位置移动到另一个位置，直

至完成。这种类型的一个典型软件过程是瀑布式开发过程，其中项目从分析到设计再到实现。在 RUP（Rational Unified Process，Rational 统一过程）中，项目从初始到细化到构造再到交付。

本书描绘的内核把软件开发工作表示成由各种组件及相互的关系组成的一种连续运作的抽象机制。在这种机制中，项目不用如同前述装配生产线所描述的那样从一个位置到另一个位置。恰恰相反，当问题被转换为需求，然后转为代码、测试和部署时，有一个连续流贯穿这种机制。

这种机制的状态通过一组名为“阿尔法”（alpha）的关键指标显示出来，这组指标代表了基础组件运行有多好。这些指标通过开发团队采取的被用来响应当前状态的一系列行动，而从一个状态推进到下一个状态。

随着项目的推进，环境会改变，客户的需求会变化，团队会发展，机制会变得失衡，团队将不得不采取进一步的行动，将机制调整到正确的运作状态上来。

与装配生产线形成对比的是，这种连续机制的比喻是由敏捷的世界观驱动的。敏捷项目不会分段进行，而是通过将客户需求连续转换成软件解决方案进行。但是敏捷项目可能失衡，它们可能进入一种它们无法充分重构，或者结对过多，或者评估不可靠，或者客户不参与的模式中。

本书的内核描绘了这组允许故障被检测出并修正的关键指标和行动。为了保持机制平稳且可预见地运行，各个团队可以通过这个内核调整他们各自的行为、沟通、工作流和工作产品。

本书的中心主题非常棒。阿尔法、状态、行动等概念简单、迷人且给人印象深刻。这正是一个内核的正确观点。我想就是这个观点可以帮助整个软件社区。

xviii

如果你是一个对软件过程和工程深感兴趣的人，如果你是一位需要保持开发团队就像一部平滑机器一样运行的管理者或团队领导，如果你是一位正在寻找某种科学来帮助你了解你的开发团队的 CTO，那么我想你将会发现这本书非常有趣。

读完这本书以后，我发现自己很想手中拿这样一副牌，让我能逐一查看它们并迫不及待地玩上一把。

Robert Martin (罗伯特·马丁)

(也称为 Bob 大叔)

2012 年 2 月

软件项目四处寻找方法学，却没有找到。幸运的是，它们确实找到了一些适合它们的个别实践；但当确定一组相关的实践是否能够从开始到结束指导一个项目时，它们往往会面对相对它们的需要太过严格的教条式的概要。一种方法应该能适用于每个项目的特殊情况：它应该得到健壮的、客观的参数支撑；而且应该使得跟踪效益变得可能。

Ivar Jacobson 和他的同事们的工作，作为 SEMAT 倡议的一部分，已经采用了一种系统化的方法来确定一种经得起时间和认知检验的软件工程原理与实践的“内核”。在这种理论成就的基础上，他们用“状态”和“阿尔法”描述了软件项目的进展。项目主管和项目成员有必要在每一时刻都知道项目的当前状态。然而，这种全局状态是由系统许多不同组件的状态组成的。术语“阿尔法”包含了这种单独的组件。一个阿尔法可以是一个软件工件（artifact），就好像软件需求或代码；它也可以是一个人的元素，例如软件团队；它还可以是一个完全抽象的东西，例如机

会，机会可以导致一个项目设立的想法。每个阿尔法在一个特定的时刻都有一个状态；组合所有这些阿尔法的状态就定义了整个项目的状态。恰当的项目管理和取得项目成功需要知道每一个开发阶段的这种状态。

xxi

内核的作用旨在识别软件开发中的关键阿尔法，并且对每一个阿尔法鉴别可以演变的标准状态。例如，一个“机会”（opportunity）可以经历以下6个状态：“已识别”（Identified），“需要解决方案”（Solution Needed），“价值建立”（Value Established），“切实可行的”（Viable），“得到处理和满足”（Addressed），“利益增加”（Benefits Accrued）。其他阿尔法具有类似的标准化的状态集。

本书的主要价值在于确立这些基本阿尔法和它们的状态，建立一种工程学方法，通过一个标准化的控制集，对于项目所处的阶段有一个明确的观察。

这种方法是开放的，因为它没有描述任何特定的实践，反而使得整合许多不同的实践成为可能，这些实践不需要非得来自相同的方法来源——例如某种敏捷方法的变体——但可以从不同的来源中结合出好的思想。许多案例研究阐明了如何在实践中应用这些思想。

软件从业者和软件工程教师对于基础牢固的方法论工作有迫切的需求。本书为任何有兴趣将软件项目开发转变为具有合理工程基础的学科训练的人提供了一个坚实的基础。

Bertrand Meyer（贝特朗·梅耶）

xxii

2012年3月

软件驱动着我们的世界；软件密集型系统（*software-intensive systems*），如同 Grady Booch 所称呼的一样，是驱动资本及衍生品交易、通信、物流、政府服务、大型国内和国际军事组织管理、医疗系统的核心架构，甚至允许小学教师 Smith 先生给小学生 Susie 发送家庭作业。甚至机械系统也让位于软件驱动系统（例如，想象一下遥控飞机）；这种趋势不是在减速，而是在加速。我们依赖软件，并且我们经常依赖它维持我们的生活。令人惊奇的是，通常，软件开发相比一门工程学科而言更加如同艺术家的作品。

当参观埃及或希腊的伟大的古代庙宇的时候，你是否曾经惊奇那些古代的建筑师或建筑商是如何知道怎样建一个经得起时间考验，历经成百甚至上千年风霜以及地震、战争和天气洗礼依然幸存于世的伟大结构？埃及人在那个时代具有令人惊讶的数学能力，但是三角测量几乎是他们的最高技术成就。事实上，位于现在土耳其的塞尔库克以弗所的塞尔苏斯图书馆伟大外观的幸存，更多的是运气，而不是古代人对于建筑理解的某种惊人能力。

当然，现在不是这种情况。建筑现在被称作土木工程，而土木工程是一门工程学科。没有人会考虑回到那个手工设计、手工建造且具有更加危险结构的远古时代。虽然今天的某些建筑仍然无法应对强烈的天气现象，但绝不是 500 年前那样高的比例。

多么自相矛盾！在一些大型复杂的系统的设计中我们依赖某种明确的工程方法学，而在其他一些大型复杂的系统的开发中，我们相当地满足于依赖某种特别的、工匠式的手工工作。可以肯定的是，不能总是这样做；通常，更加严格的流程与分析方法被用到“不允许”失败的软件密集型系统软件的开发中，这需要更多的时间和资金投入；机载航空电子设备和其他嵌入式系统设计经常远比桌面计算软件更加苛刻且昂贵。

事实上，这更大程度上是衡量计算领域比其他任何领域更加年轻的一种方法，而且我们领域的年轻绝没有什么比缺乏一种用来支撑软件开发过程的统一理论更加明显。当我们没有一种定义了软件开发过程的全球共享语言的时候，我们如何能够期望计算领域保持始终如一的软件开发过程，始终如一地在全球大学授课，始终如一地被软件开发组织支持，始终如一地被软件开发团队交付？

然而，值得注意的是，就像不止有一种方法建造一座建筑，同样不止有一种方法构造软件。所以我们需要的语言或语言簇应该定义夸克和原子而不是分子——因为原子和亚原子部分可以让我们混合和匹配，从而定义、执行、度量和改善软件开发过程本身。我们可以期待未来几年中软件开发领域关于敏捷开发与非敏捷开发之间，传统团队成员编程与结对编程之间的继续竞争；但是我们应该需要且期待我们选择的过程模块可以被持续地应用，并在必要时进行配套和比较，以及度量其功效。这个核心过程设计语言被称为本质（*Essence*）。注意，事实上本书有一个由设计基本实体组成的“内核”，这些基本实体用一种公共语言定义；我把这个困惑留给本书作者后续做出具体解答。

在 2009 年下半年，Ivar Jacobson、Bertrand Meyer 和我在一起试图定义一个能被广泛接受的过程设计内核和语言的需求，并且组建了一个国际团队解决这个需求。我们三个人在软件领域中拥有完全不同的背景，但都曾长期奋战在编

写代码的第一线，都曾领导过软件开发团队，并且我们都尝试通过各种不同的方式解决软件复杂性问题。我们的类比有所不同（歌剧方式的类比显然是 Meyer 教授的），我们的团队领导风格有所不同，并且我们的出发点显著不同。然而，这些差异促成了一次被称作软件工程方法和理论（或者 SEMAT）的国际合作。一个本质内核、一个 OMG 主要标准过程和这本书都是这次合作项目的成果。

在我们周围，形成了一个包含了一群伟大思想家的杰出团队，这个团队两年前在苏黎世联邦理工学院召开了第一次会议，紧接着很快又开了一系列其他会议。团队努力将不同的经验和世界观融合在一起，构建一个软件工程的内核——一个如同原子可以混合及配对组成各种各样的分子一样的内核，这个内核可以按需连接，可以在蓝图设计中呈现，可以用于分析，可以在实践中用于定义、雇用、管理和度量实际开发团队。在我写这个序言的同时，OMG 正在考虑如何将这个团队的工作采纳为一个国际软件开发标准。这对于软件领域来说是一个激动人心的时刻，如同我们从不同的工匠队伍偶尔一起高效工作，转变到工程师们使用良好定义的、可度量的、标准化的以及一致构建的实践来构建软件一样。

xxv

软件开发产业需要一个用来定义软件开发实践的内核及语言，这些实践可以被混合及配对，可以被从其他组织引进，可以被度量，可以被集成，可以被比较和对照它们的速度、质量及价格。很快我们将停止通过手工交付软件，而且我们伟大的软件大厦将停止倒下。SEMAT 及其本质可能不是软件工程发展的终点，并且它当然也不是第一次这样的尝试；但是它们提供了一个在软件领域中，发布被广泛接受的工程学的重要机遇。

这本充满思想的书提供了一个思考问题的良好基础，以及一种解决该需求的语言，每个软件工程师都应该阅读这本书。

Richard Mark Soley, Ph. D. (理查德·马克·索雷 博士)

写于太平洋上 38 000 英尺之高空

2012 年 3 月

xxvi

每个研发软件的人都知道这是一个复杂且具有风险的业务，并且总是对能够帮助他们更好开发软件的新思想保持警惕。幸运的是，软件工程仍然是一个年轻且处于成长中的行业；一个每年在最佳实践中见证新的创新和改进的行业。只需要看看已被软件开发团队广泛采用的精益和敏捷的思想所带来的改进和效益，就可以体会到这些新思想对于我们软件产业发展是必不可少的。

成功的软件研发团队需要在快速交付可工作的软件系统、满足各利益相关者的要求、控制他们的风险以及改进他们的工作方法之间寻求一个平衡。为此，他们需要一种有效的思维框架——一种能在他们目前工作方法和任何其他他们想要采纳的新思想之间建立桥梁的思维框架。本书以一种可以付诸实践的内核方式介绍了这种思维框架——我们相信这种思维框架将会对希望平衡他们的风险以及改进他们的工作方法的任何团队带来帮助。

启发

本书受“软件工程方法和理论 SEMAT 行动呼吁书”