

吃掉IT大象

从绿海到棕海

(英) Richard Hopkins 著
Kevin Jenkins

盛海艳 贺西玲 赵俐 等译

用于管理、演进和转换遗留IT系统的实用的、完整的方法
适用于每一位IT执行官、经理、架构师、程序领导者、项目领导者和主管分析师

Grady Booch IBM院士

寇卫东 IBM大中华区总工程师

严成文 IBM中国软件开发中心Rational总经理

欧阳璟 《程序员》杂志编辑部副主任

联合鼎力推荐



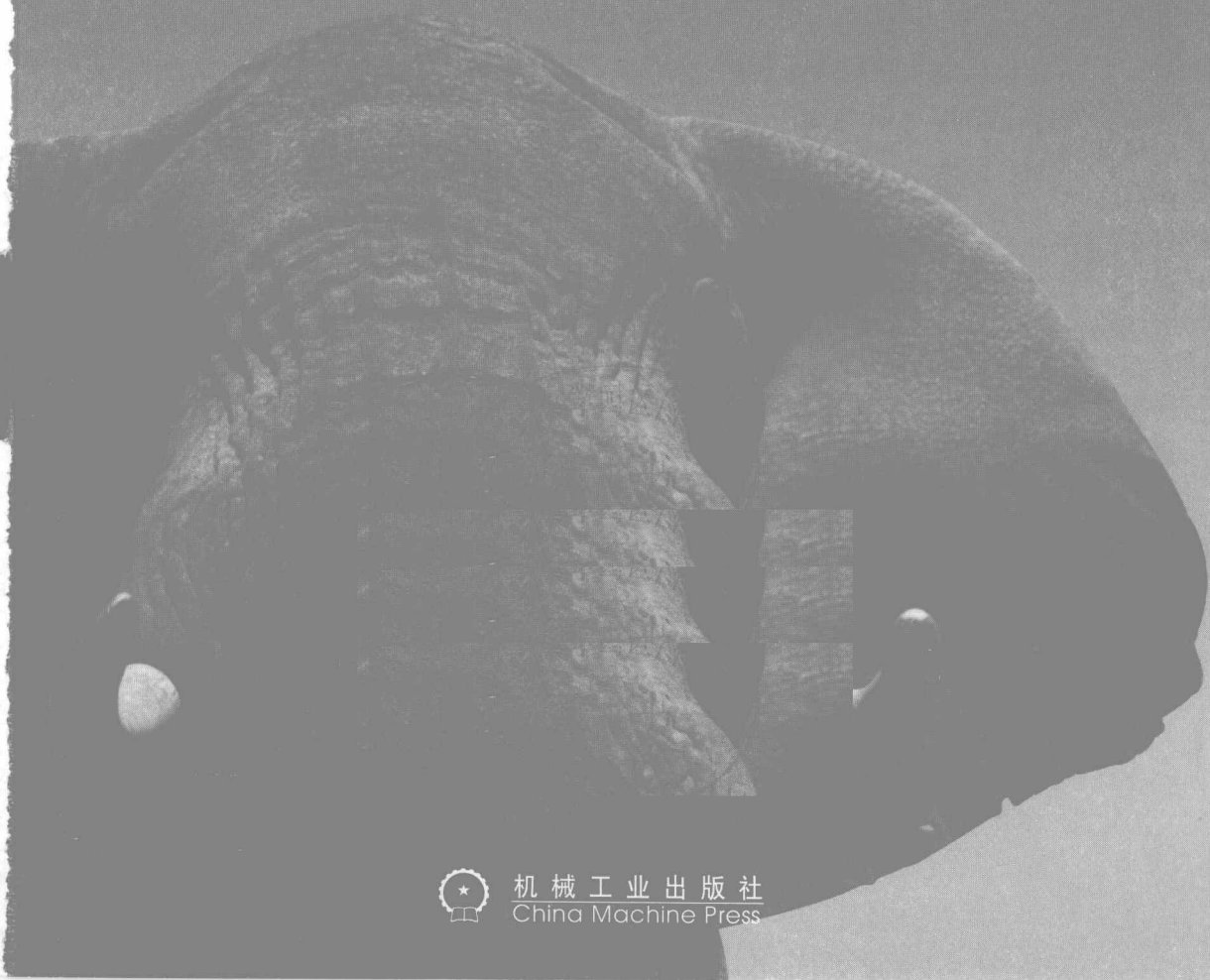
全球IT行业经典
《谁说大象不能跳舞》
姊妹篇



机械工业出版社
China Machine Press

吃掉IT大象

从绿海到棕海



机械工业出版社
China Machine Press

本书是一本适用于大型IT项目管理人员的图书, 重点介绍一种全新的项目开发方法: 棕海方法。本书立意新颖, 语言生动, 书中穿插大量真实案例, 以方便读者的理解。书中解释了为什么日积月累的业务和IT复杂性是大型项目失败的根本原因, 并展示了如何通过“一口一口吃掉大象”来克服这种复杂性。借助此书, 我们将学会如何管理棕海项目的每个阶段, 如何利用突破性的协作、沟通和虚拟工具, 包括Web 2.0、语义软件工程、模型驱动的开发和体系结构, 甚至是虚拟世界。

Authorized translation from the English language edition, entitled Eating the IT Elephant: Moving from Greenfield Development to Brownfield, 9780137130122 by Richard Hopkins; Kevin Jenkins, published by Pearson Education, Inc, publishing as IBM Press, Copyright © 2008 International Business Machines Corporation.

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

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and CHINA MACHINE PRESS Copyright © 2009.

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。

版权所有, 侵权必究。

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: 01-2009-1731

图书在版编目(CIP)数据

吃掉IT大象: 从绿海到棕海 / (英) 霍普金斯 (Hopkins, R.) 等著; 盛海艳等译. —北京: 机械工业出版社, 2009.3

(IT文化系列丛书)

书名原文: Eating the IT Elephant: Moving from Greenfield Development to Brownfield

ISBN 978-7-111-25521-5

I. 吃… II. ① 霍… ② 盛… III. 项目管理—应用软件 IV. F224.5-39

中国版本图书馆CIP数据核字 (2008) 第172800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: 盛东亮

北京京北印刷有限公司印刷

2009年3月第1版第1次印刷

170 × 242mm · 11.5印张

标准书号: ISBN 978-7-111-25521-5

定价: 39.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

推荐序一



一代伟人毛泽东曾说“一张白纸，没有负担，好写最新最美的文字，好画最新最美的画图”。如果纸上已经有部分字、有部分画，怎样写，怎样画，才能使我们的思想跃然纸上？对这个问题，少有人问及，方法更是无从谈及。

从零开始、无牵无挂的环境对软件开发来说也是最理想的。然而，世界上大多数软件开发环境都不是理想的。企业、政府、机关等经过多年的积累建设，或多或少都会有一些IT应用。面对新的需求和已有的IT环境，我们应该如何进行软件开发？这是一个非常现实的问题。令人遗憾的是我们在学校里学到的基本上都是理想的从零开始的软件开发。我们要回答这个问题就必须自己在软件开发实践中不断摸索、不断犯错、不断总结、不断学习。其时间成本、软硬件成本、开发资金等各种资源成本都是非常高的。现在，令人高兴的是我们不必再支付这样高昂的成本了，因为我们有了这本专门针对此问题的著作。

来自英国的两位IBM资深专家 Richard Hopkins 和Kevin Jenkins写出了这本值得一读的好书。这两位专家都是技术精英，一位是工程与技术学会的会士(Fellow)，而另一位则是英国计算机学会的会士。Richard 是一个很有创新意识的人，他在上大学时就开发出了能写诗的软件。此软件写出的诗居然骗倒了英国语言学的教授。Kevin是一位数学天才，他能将三维流体的数学软件建模玩于掌中。两位都有很多年的软件实践经验。Richard Hopkins 目前还担任IBM英国服务部门的副合伙人。当他们知道我为他们的著作写中文序时，都非常欣喜，并且深表谢意。他们很乐意看到他们在本书中提出的方法论能得到中国软件界的认识和实践。

在二十多年的软件职业生涯中，我读过很多软件方面的书。我认为这本著作非常有特色。首先，这本书提出了一个有根本意义的问题，就是我们面对的不是从零开始、无牵无挂的软件环境，而是即有现有的IT应用或者系统，又有新的需求，我们应该如何进行软件开发？其次，作者们不但提出了问题，而且给出了一套解决问题的方法论。这已经是非常可贵的了。更难能可贵的是他们把这些方法论付诸于实践，并且取得了可喜的成功。

如果读者能仔细研读本书，学习书中的方法论，把这些方法应用到相应的软件开发中，我相信读者就会有收获。读书如品茶，只有反复品味，你才能知其味。只有知其味，你才能识其价。

寇卫东

IBM 软件集团两岸三地大中华区 总工程师



推荐序二

正像Grady所说，绿海开发是一种巨大的乐趣。从大学毕业到我计算机软件生涯的第一个十年中，在以计算机软件起家的公司里应用绿海开发给我带来了许多乐趣。然而，当Grady和我从其创办时起就共同效力的公司在2001年被Rational收购之后，Grady所描述的绿海开发带来的乐趣也就停止了。在我能够游刃有余地运用棕海开发（就是2003年）之前，Rational就被一只能够跳舞的蓝色大象——IBM收购了。是IBM让我领略到了另外一种乐趣，那就是应用一种由Richard和Kevin命名的棕海开发方法。

无论在帮助IT业务人员解决他们棕海开发中遇到的挑战方面，还是在解决自身的棕海开发挑战方面，IBM都是创新的领导者。例如，IBM Rational的企业现代化目标是为企业IT提供有效的工具来应用SOA解决在棕海开发中遇到的挑战，同时，这些工具也应用在IBM内部来对其大型机上的软件资产进行改造。

从改革开放到中国经济高速发展和腾飞的数十年来，建筑、道路、通信线路等基础设施不断重构和重建以适应连续翻番的、超乎预料的需求，这是我们亲眼目睹并且印象深刻的事实。而我们所未能亲眼目睹和触及的是，中国的IT产业也在经历类似的过程。可想而知，有效的棕海开发会给企业和社会带来多么大的效益。

本书从一个全新的视角提供了一个棕海生命周期的、有效的解决方案。Richard和Kevin不仅教会了我们如何吃掉IT大象，更重要的是，他们启发我们思考如何避免培育出难以吃掉的IT大象，而是要培育出能够跳舞的大象。

中国经济的快速发展带动了中国IT产业的迅速成长，每天都会出现更多的中国IT大象，并且它们在日渐壮大。本书中文版的出版是一件幸事，一定能够帮助中国的企业吃掉更多的不会跳舞的IT大象。

严成文

中国软件开发中心Rational 总经理

2008年末



信息技术（IT）是一门神奇的学科，自悄然诞生的那一刻起，就以势不可挡的脚步走入我们的生活，它凭借神秘的力量潜移默化地改变着我们生活的点点滴滴，也改变着整个世界。

然而，在人们惊叹于信息技术的巨大创造力的同时，不应忘记这样一个事实：几乎70%的真正意义上的大型IT项目以失败告终。这些项目要么超过了交付期限，要么成本超支，甚至有些项目尚未完成就被迫取消了。导致IT项目失败的因素有很多，包括全球化因素、组织和规划问题、变更管理不完善、对环境复杂性的理解不够、需求定义不完整等等。本书的目标就是解决这些问题。

本书介绍了一种审视IT项目的全新方式，并引入了一种新的方法——棕海方法。棕海方法的概念来自于建筑业中的“棕地开发”，即现有城市、郊区和乡村土地的重新开发。在IT领域中，棕海开发是指在现有环境中对遗留系统进行重新开发，亦称为重构，由于现有环境可能是被各种复杂性“污染”的，因此棕海开发变得更加困难。与棕海开发相对的是绿海开发，即全新的IT项目的开发。目前我们所使用的大部分IT项目开发方法都是绿海方法，而大多数的项目却是棕海项目，从一张白纸开始的绿海项目已经非常罕见了。为了让大型的IT项目摆脱失败的命运，我们必须从绿海开发迁移到棕海开发。

本书两位作者均来自IBM英国服务部门的IT架构师，他们曾负责IBM的一些大型系统的集成和重构项目，有非常丰富的经验。他们的棕海方法已经在一些IT项目中卓见成效。当作者一步一步揭开棕海方法的面纱时，一幅生动的画卷就展现在我们面前，我们看到了一种全然不同的思想，这种思想是一枚能够点燃无数灵感的火花。

本书共分两部分。第一部分适用于所有读者，它指出了大型IT项目一贯采用的错误思想，并确定了失败的根源。第二部分解释了棕海的技术和实践两个方面。

本书立意新颖，论述详尽，特别适用于CIO、CTO、IT主管、项目执行官、项目主管、首席架构师或主管设计师。那些对成功交付IT项目感兴趣的技术人员也将从本书中获得大量有益思想和创意。

参与翻译和校对工作的人员有盛海艳、贺西玲、赵俐、马燕新、王善凤、王举华、盛立良、于波、王玲、张延青等。赵俐完成了本书的统稿和审校工作。

衷心感谢机械工业出版社华章分社陈冀康先生在翻译工作中给予的精心指导和宝贵意见，同时感谢华章分社编辑所做的大量工作，使本书得以顺利、快捷地出版。由于时间仓促，译者水平有限，译文难免会有一些错误或不妥之处，恳请读者批评指正。

译者

2008年9月



粗略地计算一下，我们就可以知道，全球每年约产生330亿行新代码或修改后的代码。日积月累，这意味着自20世纪40年代和50年代以来（当更高级的编程语言开始流行时），我们已经生产了约1万亿行源代码。

一方面，这种规模的产量说明我们的行业是一个充满活力和创新的行业。另一方面，这是一个令我们惭愧的事实，通过这1万亿行完全由不同的人手工编写的代码，我们由此改变了整个世界。

事实上，在每年产生的330亿行代码中，为数不少的一部分刚刚诞生就已经死亡，或者很快就被丢弃了。不过，很多代码有着较长的“半衰期”，有些代码的生存时间甚至达到10年、20年、30年或更长的时间。对于很多开发人员来说，他们今天编写的代码到了明天就成为遗留代码，某一天，他们的下一代或再下一代可能会注视着这些代码，尝试使用它、适应它、改进它，并提出这样的问题：“这位开发者到底在想什么？”

坦白地讲，绿海开发是一种巨大的乐趣，因为我们从一张白纸开始，不受过去任何事情的羁绊。在很大程度上，我们在学校中讲授的是绿海开发；此外，新公司看起来比老公司敏捷得多，因为他们没有遗留系统的瓶颈问题。困境将降临到那些刚刚步入现实世界的学生们头上（现实世界与大学不同，除非他们从大学的温床直接进入新成立的公司的温床），困境也会降临到那些开始步入持续性开发并迅速认识到已经无法从头开始的公司身上。

Richard和Kevin为我们揭示了一个常常被业界忽略的现实：即不断演变的遗留系统的问题，他们将其称为“棕海开发”。在当今的经济效益模式下，典型的系统都在不断演进（我们无法阻止这种演进），同时也在不断增长。作者认为问题的根源在于复杂性，并提供了一种聚焦于基本抽象和有效沟通的新方法，从而一步一步地解决转换问题。他们的视图、资料库、转换和工件模式提供了用于推理和执行棕海系统转换的一种方法。他们提出了一个棕海生命周期，包括调查、工程、验收和部署，从而提供了控制转换的手段。

正如一句谚语所说“大象是要一口一口吃掉的”。Richard和Kevin带领我们来到摆好刀叉和其他工具的餐桌旁，为我们展示了一种在房间里吃掉大象的方法。

Grady Booch

IBM院士

2008年1月



序 二

1969年，我从学校直接迈入计算机行业，成为是一名计算机工程师。在近40年的职业生涯中，我主要致力于应用程序开发和系统集成领域。我在1969年编写了第一个应用程序，它是一个计算机辅助设计（CAD）的图形应用程序，帮助硬件工程师设计印刷电路板。此应用程序为电路板设计人员提供了一个工具，它具有电子元件的必要物理规则，以及如何使用的规则。在20世纪70年代初，我开发了CAD和其他应用程序，用来帮助建筑师设计大型公共建筑，例如学校和医院。这些系统在建筑师和市政工程师的建筑设计过程中提供帮助；通过捕获设计，它能够生成所有必需的图纸和建筑材料清单。

在这40年当中，我经历过各种不同的角色，包括程序员、分析师、设计人员、架构师、项目经理和问题解决专家。我所开发的系统用于广泛的行业，包括制造业、银行业、保险业、零售业、公用事业以及地方和联邦政府。现在，我是IBM全球业务服务部的IBM院士¹，也是IBM技术研究院的积极成员²。我的主要职责是从技术上规划和确保大型、复杂的系统集成和战略外包计划及投标的技术健康。我是一名皇家特许IT专家（CITP）、皇家特许工程师（CEng）、英国计算机协会会员（FBCS）³，和英国工程技术学会会员（FIET）⁴。

回顾20世纪70年代我们在电子电路设计以及建筑业中的尝试，我对于IT行业在基于工程的方法的采用方面深感失望，甚至在某种程度上感到绝望，因为IT行业在规划、设计、构建、集成和测试IT系统时，缺乏对基于工程的方法的成功采用（这种方法应该由基于计算机的工具来支持）。在当今世界中，在没有工程原则和IT行业提供的工程工具的情况下，要想开发复杂系统（例如Airbus 380）是不可想象的。IT行业在采用工程技术开发复杂系统方面，还相当不成熟。IT行业已经无法再依靠相对不成熟的实践，这些实践通常是由办公生产力工具提供支持的，例如文字处理工具、幻灯片工具和电子表格。IT行业需要更广泛地采用真正基于工程的技术，这些技术由专门为工程师设计的工具提供支持。

根据我近年来的经验，构建定制（自定义）应用程序或定制商用成品（COTS）软件包的总成本和复杂度正在不断增加，风险也随之加大。通过进一步的调查，可以得到一个明显的事实，即并不是构建成本增加了，而是在系统环境中集成这样项目的规模和复杂度增加了。从我最近的经验来看，新的构建工作与集成工作

的比例为3:1。对于在新功能上花费的每一美元，将该功能转化为生产的总成本是4美元。此成本并不包括最终用户的培训费用。在系统规模和复杂度不断增加的环境中，维护成本也将增加。此外，企业还必须满足不断提高的法律和法规要求。这些因素导致用于新开发的预算下降，并且在全球化24×7的服务文化中部署新功能的机会也降低了。IT创新被遏止了。现今使用的方法和工具（虽然受限）主要是针对绿海系统环境的。现实情况是，21世纪的大多数组织都拥有一个现有的、复杂的系统环境。我这里所说的系统环境（systems landscape）同时包括业务环境和为其提供支持的IT系统。这些IT系统又由应用程序组成，并且它们的数据通常部署在复杂的网络和计算机基础设施上。这些系统的文档化程度往往很低，并且后续的维护高度依赖于少数知识丰富的“系统专家”⁵。IT行业需要一种更强的结构化方法来理解这些系统环境。

这就是当今世界的现实状况，本书作者Richard Hopkins和Kevin Jenkins还有我，都在客户的现有复杂系统环境中规划、设计和实现新的系统。现在是IT行业面对现实状况的时候了，我们需要新的开发方法和工具来解决这些问题，并将我们的行业带入21世纪。

解决此问题的重要的第一步是提供一个同时用于描述问题及其解决方案的名称。在命名过程中，两位作者将目光转向建筑行业，其中越来越多的新建筑是在棕地⁶上开发的。这是对当今大部分在棕海系统环境中开发的新系统的类比；根据我的经验，超过90%的新开发是部署到棕海环境中的。挑战并不仅限于遗留系统的转换，还包括将其集成到棕海系统环境中。

本书描述了一种用于未来系统开发的全新方法。它是一种已经认识到这些挑战的结构化方法，它基于工程原则，并且由适当的工具提供支持。它是专门为应对棕海开发挑战而设计的方法。

Chris Winter

CEng CITP FBCS FIET, IBM院士

IBM技术研究院成员

注释

1. "IBM Appoints Six New Fellows Who Explore the Boundaries of Technology."
<http://www-03.ibm.com/press/us/en/pressrelease/21554.wss>, 2007年5月。
2. IBM Academy. <http://www-03.ibm.com/ibm/academy/index.html>。
3. British Computer Society. <http://www.bcs.org/>。
4. The Institution of Engineering and Technology. <http://www.theiet.org/>。

5. Lindeque, P. "Why do large IT programmes fail?" [http:// www. ingenia.org. uk/ ingenia/ articles. aspx?Index=390](http://www.ingenia.org.uk/ingenia/articles.aspx?Index=390), 2006年9月。
6. 美国房地产经济人协会 (National Association of Realtors®) 将棕地描述为“现有的城市、郊区和乡村土地的重新开发，它们已经由基础设施提供服务，这里的基础设施包括已经或者可能被污染的‘棕地’地点，这些开发可以刺激发展并提高社会的经济活力。现有环境中的开发是实现经济有效增长的一种方法，同时能够为居民提供更多工作机会、公共服务和生活设施。”
-



每家企业的愿望都是为满足客户要求而对自身做出迅速调整。这样的调整通常涉及IT支持系统的改变。当需要进行重大的业务变更时，伴随的IT变更也将非常巨大。但是，这些大型项目常常遇到问题、超出预算、延期或简单地被取消了。在2006年，65%的IT项目以失败告终¹。大型项目的成功率甚至更低。当代价非常高昂时，这样低的成功率令人担忧。本书指出了IT行业当前方法的最根本的核心问题，并提供了一种全新的未来方法。大规模的业务和IT变更中涉及的所有人员都应该阅读本书。

大象的生日

IT行业有很多重要事件，但1964年IBM新一代大型机（称为System/360）的引入标志着一个全新时代的开始。在此之前，购买新的商用计算机意味着必须重写现有软件。System/360改变了这一切，它引入了一个兼容的计算机家族以及相关设备。在某台计算机上运行的程序也可以运行在任何其他计算机上。业界纷纷以类似的产品效仿，一举改变了IT的性质。

现在，IT投资很容易得到维护。运行在System/360上的程序仍运行在现今的IBM大型机平台上。

最初，这是一个不易察觉的变化，但它是一个重大的里程碑。从那时开始，IT复杂性开始在企业内部累积。系统随着企业增长。数以千计的“人年”（person-year）时间、工作和资金流入这些IT系统中。它们日趋复杂，它们变成了大象。

同时，各种IT风格也经历了兴衰演变。多年以来，最初的结构化程序得到了面向对象编程的补充，经过了基础组件开发的包装，并且通过面向服务架构（SOA）得以传播。这些运动中的每一个演变都有着自己的解决复杂性的策略，但它们都没有抓住问题的核心。

当今的IT系统如此复杂，以至于要了解它的架构都很困难，当我们试图靠近它时，它却迅速地溜走了。维护它的责任分配到拥有各种技能的小组中，而且数不清的产品和程序共存，用以维系企业的功能。为了处理这个“九头怪”，我们绘制了高层体系结构图，以便使事实看起来更简单。但这些图只是一种幻想，一种假象，一种外观。它们至少近似于一种简单的假定和高层的沟通。在最坏的情况下，它们灌输了一种盲目的乐观主义，使人们对改变这种复杂性的能力产生了错

误认识。

这样的“云雾”图无法永远隐藏真实的复杂性。为了实现业务目标并改变这些系统，我们必须理解、沟通和治理真实的复杂性。没有人能够理解所有复杂性，因此完成这些任务就需要大量良好协调的团队工作和明确的沟通。高度的复杂性与数以百计的不同人员之间的沟通需求交织到一起，会成为摧毁大型项目的杀手。

是否需要从绿海迁移到棕海

一般情况下，现在IT系统已经不再是在绿海上实现的。1964年以来累积的复杂性意味着大多数IT项目的环境都是一个巨大的挑战，而且与几乎无数的环境约束交织在一起。

这是大多数大型IT项目失败的根本原因。只有30%的大型IT项目获得成功。

大型项目通常是在“被污染的”环境中执行的，在这样的环境中，我们需要仔细考虑在哪里以及如何构建项目；某个位置上发生的改变可能会以无法预料的方式影响其他系统。在这样的环境中，棕海性质多于绿海性质，而且IT行业需要采用面向棕海的方法来成功解决这些问题。

本书引入了这样一种棕海方法，并解释了为什么目前我们所使用的方法本质上仍然是绿海方法。本书的读者对象是那些想要改变企业的人，他们知道只有在过去的基础上构建项目才能实现所需的改变。本书适用于以下情况：

- 正在构思对IT环境做出重大改变的CIO、CTO、IT主管、项目执行官、项目主管、首席架构师或主管设计师。
- 无法承担替换整个IT环境的费用。
- 系统与大量不在控制范围之外的其他系统进行通信。
- 打算重构现有IT环境，以保持未来的灵活性。
- 对当前的IT项目失败率感到深深失望。
- 正打算将很大一部分IT开发和测试工作外包到国外。

作为本书的作者，我们是IBM全职的执行IT架构师，我们接触过上述各种情况，曾经负责IBM的一些最大型的系统集成和重构项目的技术方向和日常实现。我们坚信现有的绿海开发方法在通过IT解决方案解决当今业务问题方面日渐失效。坦率地讲，多年以来我们积累了宝贵的经验，我们不遗余力地解决IT交付的问题。近年来，我们刻意去寻找一种不同的方法；以下几页详细介绍了我们与同事共同努力所获得的成果。我们可能是信奉非正统见解的人，但我们也是实用主义者，我们在这里分享的见解已经极大地加速和简化了大量的IBM项目。

我们从未认为大型IT项目的高失败率对行业有任何利益，也不是通过高失败率这个事实来宣传我们的方法。如果我们能够帮助减轻不可避免的复杂IT环境的影响，并破除大型项目的某些沟通障碍，那么我们相信成功率将会提高。

读者摘要

本书不是一本技术手册，也不是“食谱”；它不包含任何代码，而且我们最大限度地减少技术图形和术语的使用。本书旨在改变处理大型和复杂业务及IT重构项目的方式。

为了让尽可能多的人更容易阅读本书，全书分为两个部分。

第一部分适用于所有读者，这一部分首先定义了对于大型IT项目来说什么是错误的，并确定了失败的根源（第1章和第2章）。本书的核心（第3章和第4章）重点定义了一种替代解决方案——Elephant Eater以及配套的棕海方法。在第5章中，我们将看一下作为结果出现的新的业务种类。

第二部分为那些希望实现这种方法的人解释了棕海的技术和实践两个方面。这一部分首先分析了现有Elephant Eating技术（第6章），并解释了为什么棕海是不同的（第7章）。在第8章和第9章中，我们研究了Elephant Eater的内部构造以及一些用于实现它的新技术。本书以解释如何在项目上实现棕海方法及其获益作为结尾（第10章）。

对于决策者来说，已经发布的大量技术信息将帮助任何组织以自己的方式来采用棕海方法的核心技术（或等效的技术），我们正是利用了这些技术来实现棕海开发的（参见第8章和第9章的注释）。我们希望本书的核心是通过一种全新的项目方法（而不是技术）来实现业务和IT变更。

第一部分：棕海简介

第1章“吃掉大象是一件难事”引入了一个比喻：执行大型IT项目可以比作吃掉一头大象。该章考查了大型项目为什么会失败，并提供了一些有关如何克服常见失败原因的最佳实践。

第2章“语言的混淆”解释了为什么这种累积的IT复杂性是失败的根源，该章重点关注IT复杂性所产生的人员沟通问题。它还专门考查了业务与IT之间的“巨大鸿沟”，这使得问题更为复杂。

第3章“我们需要一个大嘴超人”介绍了棕海的核心概念。该章考查了如何实现棕海，以创建一个有效的Elephant Eater机器。

第4章“通向大脑的高速公路”。我们对于IT人员无法像其他类似的专业人员

(例如实际的建筑师)那样进行有效且高效的沟通感到绝望。第4章描述了如何将棕海方法与VITA体系结构结合起来,从而开创新形式的沟通、远程协作和复杂IT问题的虚拟化。

第5章“神秘的元人”是对本书第一部分的总结,它通过考查棕海的可能影响进行了总结。该章预想了一种新的业务种类,它无疑比现今的方法更关注客户,并且更敏捷,该章还解释了如何令这样的业务成为现实。

第二部分: Elephant Eater

第6章“只有在完美的世界中,抽象才有用”。本书后半部分涉及更多技术方面,第6章是这一部分的开始,它定义了Elephant Eater的特征。该章考虑了现有的“Elephant Eating”方法,并指出由于这些方法广泛采用了分解和抽象,反而增加了项目的难度。

第7章“Elephant Eater 的进化”考查了棕海的技术和项目根基,并解释了它与先前思想的关键区别。该章以一些可能的场景和真实的项目示例结尾,棕海方法已经(或可能)为这些项目带来了获益。

第8章“棕海开发”介绍了如何在项目上部署棕海开发方法。它介绍了如何在基于敏捷的开发技术和基于瀑布的技术之间做出新的平衡,并提供了二者的一些最佳元素。该章还描述了调查、工程、验收和部署的核心阶段,并展示了方法的获益。

第9章“Elephant Eater的内部机理”。如果说第8章描述了在棕海项目上发生了什么,那么第9章解释了这些事情是如何发生的。该章深入分析了Elephant Eater的内部工作原理,并解释了它如何吃掉大象。该章通俗易懂,还可以作为新的语义技术的入门,新的语义技术为Web 2.0和语义Web提供了支持。

第10章“Elephant Eater 实战演习”。该章是全书的结尾,它介绍了Elephant Eater的实际应用,以及如何帮助解决当今一些最困难的IT问题。该章总结了棕海方法的关键获益。

漫游棕海

希望读者从阅读本书中获得乐趣,就像我们从写作中获得乐趣一样。如果读者希望了解更多,可访问网站www.elephanteaters.org。此外,如果读者希望看到棕海的更多生动的特质,在Second Life上有两个演示。一个Second Life站点专用于本书[Cypa 30,180,302]。另一个Second Life站点专用于IBM的棕海使用[IBM 140,150,60]。我们期待在那里与读者相遇。

致 谢

离开家庭对我们的支持和理解，本书是不可能完成的，即使在有闲暇时间的时候，我们也很少陪伴在他们身旁。此外，我们还应感谢以下人员或公司所做的贡献（按字母排序）：

- Bob Lojek——他将模型驱动的体系结构推进到新的水平，并在软件“宝贝鱼”方面做了首次合理的尝试。
- Chris Winter——一位真正令人鼓舞的IBM院士，他比我们在业界遇见的任何人都更相信社会和人们的力量。Chris是本书的关键推动者和技术顾问。
- Christian Hance——Christian是IBM的一位项目执行官，架构师们坚信他能够带领他们创建第一个棕海迭代。我们非常感谢他同意对本书进行评审。
- Cúram Software——首次为我们引入了模型驱动的开发体系结构的实例。
- Fred Brooks——他将他的重要著作《The Mythical Man Month》称为软件工程的圣经，因为所有人都听说过它，一些人读过它，但很少有人遵循它。我们希望读者同时做到这三点，并且将其当作一本“圣约书”，而不是流于某种形式。我们感谢Fred的一句名言：“棕海远远比绿海难得多，无论是对软件还是对房屋进行重新建造，都是如此。”
- Ian Hughes——也许他的“ePredator potato”^①这个名字更广为人知，Ian将我们引入到Second Life，并在基于PowerPoint的体系结构方面做出了一项重要建树（参见第4章）。
- Ian Scott——执行IT架构师实际上不应亲自从事编码工作，但Ian迈出了勇敢的一步，他秉承了专利方法和代码（即第4章中用于生成系统图的代码），此代码的功能现在是世界领先的，感谢Ian。
- IBM——本书提出了大量基本思想；IBM认为本书的出版使Gerstner的“跳舞的大象”的景象跃然纸上。
- Mandy Chessell——他将体系结构深深印入每个人的脑中，并为它申请了专利，并且无意间创造了VITA这个缩写。对了，还有他的幻灯片演示，非常好的幻灯片。
- John Tait——John提醒了我们本书的时间线。为什么我们凭直觉将问题的起源归结于35年前呢？当我们思考这个问题时，我们已经知道了答案，但此前我们完全忽略了这个问题。

① 这是Ian Hughes在Second Life中的化身的名字。——译者注

- Katherine Bull——我们对本书中的思想一直保持着热情、激情和动力，Katherine则为出版过程赋予了同样的激情。
- Kevin Ferguson——可以说，Kevin拯救了我们，只有上帝知道他为本书付出了多少时间，感谢他对我们的文字所做的细致编辑和评审。没有人会怀疑我们来自加拿大。
- Phil Tetlow——Phil是所有语义技术和复杂技术（包括Web科学）的最热忱的支持者，并且为这些工作提供了全面支持。
- 我们的客户——最后但并不是最不重要的。在这里我们不能列举客户的名字，因为这肯定会引起律师们的兴趣。但毫无疑问的是，一直推动我们前进的动力是尝试解决客户的问题，以及为客户的顾客解决问题。我们希望棕海有助于解决人们所面对的一些棘手问题。

注释

1. 1994年，根据Standish Group最初的CHAOS报告，IT项目的成功率为16%。在随后的几年中，成功率稳步提高。2006年，Standish Group报告有35%的IT项目按时、按预算交付，并且满足了用户需求。该记录中唯一的意外情况出现在2004年，这一年的失败率上升了。Standish Group的解释是，2004年有更多大项目，它们更容易失败，因为它们经常被迫放弃迭代开发技术。

2007年，Sauer、Gemino和Horner Reich所做的一份报告（与CHAOS是竞争对手）考查了412个项目。该报告发现超过65%的IT项目获得了成功，但成功的项目均不超过200人年，而本书专门针对的是超过200人年的大型项目。

Hayes, F. Chaos Is Back. www.computerworld.com/managementtopics/management/project/story/0,10801,97283,00.html.

Krigsman, M. Rearranging the Deck Chairs: IT Project Failures. <http://blogs.zdnet.com/projectfailures/?p=513>.

Rubinstein, D. Standish Group Report. www.sdtimes.com/article/story-20070301-01.html.

Sauer, C., A. Gemino, and B. Horner Reigh. "The Impact and Size and Volatility on IT Project Performance." *Communications of the ACM* 50 no. 11 (November 2007): 79-84.
