

企业架构实用指南



- ▶ 企业架构概念和方法的全面阐释
- ▶ 详解实用过程运作的系统化方法
- ▶ 包括企业架构的Rational统一过程扩展
- ▶ 创建战略蓝图的各种架构流程的统一视图

JAMES MCGOVERN · SCOTT W. AMBLER
MICHAEL E. STEVENS · JAMES LINN 著
VIKAS SHARAN · ELIAS K. JO
OLIVER SIMS 序
李琦 郭耀 译



企业架构实用指南

James McGovern, Scott W. Ambler

Michael E. Stevens, James Linn 著

Vikas Sharan, Elias K. Jo

李琦 郭耀 译

清华大学出版社

北京

Simplified Chinese edition copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

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

Original English language title: A Practical Guide to Enterprise Architecture, 1st ed. by James McGovern, Scott W. Ambler, Michael E. Stevens, James Linn, Vikas Sharan, Elias K. Jo, Copyright © 2004

EISBN: 0-13-141275-2

All Rights Reserved.

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

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

本书中文简体翻译版由 Pearson Education, Inc. 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2004-3182 号

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

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

企业架构实用指南/(美) 麦克高文(McGovern, J.) 等著; 李琦, 郭耀译. —北京: 清华大学出版社, 2005.9

书名原文: A Practical Guide to Enterprise Architecture

ISBN 7-302-11401-3

I. 企… II. ①麦… ②李… ③郭… III. 企业管理 IV. F270

中国版本图书馆 CIP 数据核字 (2005) 第 079429 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

组稿编辑: 常晓波

文稿编辑: 李 晔

印刷者: 北京市清华园胶印厂

装订者: 三河市化甲屯小学装订二厂

发行者: 新华书店总店北京发行所

开 本: 185 × 230 印张: 17 字数: 378 千字

版 次: 2005 年 9 月第 1 版 2005 年 9 月第 1 次印刷

书 号: ISBN 7-302-11401-3/TP · 7491

印 数: 1 ~ 3000

定 价: 32.00 元

致谢

若非组稿编辑 Paul Petralia 和 Peter Coad 的慧眼，本书绝无可能问世。非常感谢我们的执行编辑 Jim Markham，他花费了许多时间修订我们的原稿，精心去除语言上的瑕疵。

本书的作者小组得益于业内许多天才的构想家写下的初稿评论。这些评论来自于全世界背景不同的各个角落：从工作在大企业里的专业人员到主要院校研究机构中的教授；从信奉重型过程的到酷爱敏捷方法的；从以英语为母语的到使用各种有趣语言——阿拉伯语、汉语、法语、葡萄牙语和西班牙语的；从基层开发者到高层总裁。

我们非常感激下列人士的反馈意见(在许可的地方注明了他们的所属公司)：

- Paul Clements(高级技术员)，软件工程研究，卡内基梅隆大学
- Russ Rufer, Silicon Valley Patterns
- Charlie Mead, (总监), Healthcare Information Architecture, Oracle
- Alan Inglis(企业咨询总监), The Infrassistance Company
- Leon Kappelman(信息研究中心主任), University of North Texas
- Celia Redmore, BMC Software, Atlanta, Georgia
- Marcello Kai(IT 分析师), IBM
- Joe Schurman(独立咨询师)
- David Dossot(团队主管), Atos Origin, Luxembourg
- Spiro Lecatsas(CTO), Pitstop IT
- Taran Rampersad(软件工程师), KnowProSE.com, Trinidad & Tobago
- Jean Sirica(架构师), Hartford Financial Services
- Deklan J. Dieterly(软件开发员)
- Todd J. Johnsgard(人因工程师)
- Gerrit Muller(高级研究员), 嵌入式系统研究所, 荷兰
- Robert H. Shanahan(首席软件架构师), Northern Solutions Software
- Hubert Smits(高级咨询师), Vision Consulting, 苏格兰
- Eugen Coca(计算机体系结构教授), University Stefan cel Mare, 罗马尼亚
- Marié-Louise Dercksen(业务架构师), Comparex, 南非
- Rik Manhaeve, 比利时
- Jason Garbis(高级产品经理), IONA
- Phillip Ohme
- Alfred Smith(应用架构师), State Information Technology Agency, 南非

- Lucas Persona(计算机科学家), Widesoft Sistemas, 葡萄牙
- Sirigere Sanjeev, Siemens AG
- Charles Hawk, FedEx
- John Lamb(企业架构师), Hartford Financial Services
- Sean McBreen(.NET 架构组长), Cap Gemini Ernst & Young, United Kingdom
- Cordell Carter II(咨询管理), IBM
- Raul Abril(项目管理总监), Teradata(a division of NCR)
- Nitish Verma(企业架构师), Essense Networks Ltd, 新西兰
- G. Michael Zimmer(高级协调员), Methods and Procedures, British Columbia, 加拿大
- Harm Smit(高级项目经理), SYLLEM, Toulouse, 法国
- Marilyn Littman(教授), Nova Southeastern University, 佛罗里达州
- Alexander Schmid(企业架构师), Munich, 德国
- Rathan Dhariwal(软件架构师), Computer Horizons
- Bob Corrick(开发技能经理), Virgin Money, Norwich, 英国
- Sergio Villagra(教授), Universidad de Buenos Aires, 阿根廷
- Xijia(Frank)Chen(技术领导), Verizon Wireless, 加州
- Peter Booth(软件架构师), Major Financial Institution, 纽约市
- Dirk Muthig(部门主任), Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, 德国
- Anders Aspñäs(解决方案架构师), Fujitsu Invia OY, Turku, 芬兰
- Dr. Shane D. Shook(经理, 信息风险管理), KPMG LLP
- Gordon Tulloch(应用开发经理), Province of Manitoba, Winnipeg, 加拿大
- Lisa Crispin(经理, QA), KBToys.com, Denver, 科罗拉多州
- Huet Landry(Software Process Improvement Manager), Unisys, 弗吉尼亚州
- Thomas M. Cagley Jr.(负责人), TMConsulting Services, 俄亥俄州
- Claudio Ochoa(高级软件工程师), NEC, San Luis, 阿根廷
- Charles Poon-Teck Ling, Ph.D.(Service Provider, 架构师, 首席咨询师), Sun Microsystems Pte Ltd, 新加坡
- Nitish Verma(企业架构师), Essense Networks
- Bruce Beller(项目经理), Beller Consulting, Inc., 宾夕法尼亚州
- Horia Slusanschi(企业架构师), Unisys, Wellington, 新西兰
- Marlyn Kemper Littman(教授), Nova Southeastern University, 佛罗里达
- Paolino Madotto(经理), KPMG, 罗马, 意大利
- Mark R. Nelson(教授), Rensselaer Polytechnic Institute, Troy, 纽约州

- Sharad Patel(咨询师), KPMG, 伦敦

作者队伍广泛阅读了业界名宿的论文, 获益良多, 并把所采集到的企业架构和敏捷方法的一些想法结合到本书中。这些专家包括:

- Dana Bredemeyer
- Ruth Malan
- Douglas Barry
- Paul Harmon
- Peter Herzum
- Kent Beck
- Alistair Cockburn
- Ward Cunningham
- Martin Fowler
- Jim Highsmith
- Ron Jeffries
- Jon Kern
- Clive Finkelstein
- Steve Mellor
- Ken Schwaber
- Jeff Sutherland

James MacGovern

当别人都相信你能够写出这本书的时候, 写作会更加容易。向 Sherry——我的挚爱和灵感的一贯源泉——致以最深厚的谢意。感谢我的父母, James 和 Matie Lee, 他们抚养我长大, 一直支持着我。向我的家人: Daisy、Annika、Pamela、Demesha、Jesse 阿姨、Soogia、Kenrick、Kiley、Nicholas、Ian、Alex、Kelon 和 Keifer 致以广泛的祝福。感谢我完美的儿子, 小 James: 我将要得到你了。

感谢我在哈德福特财务服务(Hartford Financial Services)的工作伙伴, 他们从未对企业架构和本书的进展丧失兴趣。我要进一步感谢曾经阅读过我以前著作的读者们, 谢谢他们的建议、赞誉、称道我的来信和赏识。

Scott W. Ambler

我要感谢业内所有成功地在他们的组织中为企业架构采纳了敏捷方法的架构师, 以及那些业界的思想领袖们, 他们的坚持不懈让敏捷方法成为一个值得为之努力的专业。

Michael E.Stevens

在对家人的致谢中，我感谢我的妻子 Rhonda 和女儿 Katie，感谢她们在我写作本书期间给予我的鼓励和理解。我也要感谢我的母亲 Doris 和父亲 Ed 的支持。在对我的工作伙伴的致谢中，我感谢 James McGovern、Jeff Ryan、Tom Sorensen、Damon Rosenthal、Geetha Yogaratnam 以及 Houman Younessi 教授和 Charles Pelletier，感谢他们在造就我的软件开发理念过程中扮演的重要角色。最后，我感谢 Frank Hauser，他发现了我在编程方面的才华，启发我在计算机科学领域学习和成就自己的事业。

James Linn

感谢我的妻子 Terria，感谢她在我从事这个项目时亲切的宽容。我还要感谢 James McGovern 和 Michael Steaven 设立了企业架构应该是什么样的实例。

Vikas Sharan

感激众多业界的思想领袖，他们人数太多无法在这里列出，正是他们让企业架构成为一个真正的交叉学科的专业。我也感谢我的妻儿的支持与耐心。如果我没有和 Ritu 结婚，大概今生今世永远不可能写出这本书或者完成这么多的事情。当我告诉我的儿子 Shubhankar 我可能是在写一本书时候，如果他不是这么充满敬意地看着我，我大概也不会命令自己动笔。

Elias K. Jo

感谢我的技术生涯中曾经和我一起工作和共同成长的同事们和朋友们。我特别要对 Abdi Oday 说，你就是个架构师。感谢 Tony Yi，是你帮助我把业务和技术点连接起来。感谢 Brian Walter，他相信我了解我所谈论的东西。感谢 Jerry Wen，他把我第一个程序称为一个“病毒”。

也感谢我的朋友和家人，他们聆听着我的技术想法，尽管我猜他们不理解我所说的东西。感谢妈妈、爸爸、Q 和 Linda。同时，也感谢我亲爱的 Susan，让我在她那里写下许多部分(她那里比我自己的住处舒适多了)。

我也要感谢作者组的其他成员。你们的帮助和反馈在整个写作过程中极大地支持着我。

前言

从前，某位训练有素的科学家在他的实验室里工作，他把盛满试剂液的烧杯放到本生灯上，又拿起另一个烧杯，把内盛的试剂倒到前一个烧杯里。随着温度的升高，混合液的颜色开始变化，随即突然泡腾起来，散发出非常美妙的芳香。

“我找到了！”科学家欢呼着冲出实验室，把这个好消息带给他的主管们。

“我们一定要将它立即投产！”CEO说，“我们今年就能卖20亿加仑！”

于是建筑队受命修建一个200英尺高的本生灯以及220英尺高的基座来安放50万加仑的烧杯，还需要建一个500英尺高的起重机来高高举起第二个烧杯，以便把内盛液体倾倒在第一个烧杯里制成混合液。

然而，不行，这样的做法将会是愚不可及的。实验室中的试验和大规模生产相当不同。因此，企业系统若和实验室的原型系统采用同一架构，未免有点古怪，是愚不可及的。企业系统异于诸如“餐厅局域网”式的原型小系统，但是这种差异体现于架构，而不是设计思想。可是架构经常会被混淆于设计。架构表现的是在某类事物中普适的特征、结构、行为和关系。设计表现的是细节，说明某类事物中特定的个体该如何建造。

架构和设计总是存在的。但在许多时候，它们埋在程序员意识里，踪影难觅。如果现在所有的程序员都是精干的架构师/设计师，如果大家都具备长期卓有成效的企业系统开发经验，如果大家在开发企业系统或其他相关项目时都愿意和其他程序员交流思想，如果在系统开发完成后不需要其他人员做维护、甚至另起炉灶再开发一个新的企业系统，那么架构和设计难觅踪影的状况会一去不返。可惜实际情况并非如此，而是恰恰相反。

因而，架构和设计必须分开，并且明确化。架构由知识丰富的专家制造，负责沟通、启发和领导。仅有设计是不够的。企业系统的设计必须适合此类系统的外延功能需求——可伸缩性、完整性、灵活性、可建造性等——这些都是由架构决定的。

企业系统经常失败的一个重要原因是架构和设计被合并。其他一些和企业系统同样复杂的人类活动，并没有出现和这些大型IT项目相同的失败率。这是为什么？我的回答是当前IT产业在三个主要方面存在显著欠缺：

- 企业层的架构(企业架构)。
- 支持企业架构的工具。
- 支持企业架构的组织。

架构知识的燃眉之需

设计一个企业系统是困难的。它要求大量的知识和技巧。在其他产业中，专业人员在开始工作前就被授予许多必要的知识。这些产业可以说是“知识专门化”。知识被划分为各个专门的需求领域。在建筑业里，建筑师知道工厂设计和公寓设计是相当不同的，公寓设计又与教堂设计和办公楼设计不同。又如，工程师明白设计磁盘驱动器和设计飞机是相当不同的(尽管它们都会用到空气动力学)。汽车设计师了解十八轮大货车的设计和家用轿车的设计不同。所有领域都有它自己的架构，设计特定的产品要遵从它的架构。

在一个产业中，每个被关注领域以所谓的“架构方法”(architectural approach)为特征。(Richard Hubert 称之为“架构风格”，参见 *Convergent Architecture*, Wiley, 2002。)涉及不同架构方法产品的项目少有相同之处，相反地，生产含有相同架构方法产品的项目会有许多共同点。因而，项目中所使用的技术和工具也可能是相同的。设计特定的产品有据可查，有章可循，由架构方法规定其中的设计。那些跨领域(有些时候还会跨产业)的普适技术是非常重要的，但更重要的是这些技术对各种架构方法的不同应用。

按照客户需求的架构方法，制造产品所需的知识各有差别，客户经常会指定架构方法：亦即，你会听到“我需要一辆家用汽车”，而绝少听说“我需要一辆车”。

我们的产业倚重于技术，较少倚靠架构方法。显然，一个单机的图形用户界面应用程序和一个企业系统相当不同，这两者又都不同于一个工厂自动控制系统。这些应用系统都各自体现着不同的架构方法，架构方法服务于相应的那类应用系统项目，这类项目具备大量普适知识。然而，许多 IT 项目仍然由掌握着一套技术工具的专业人员着手进行，他们并不具备必需的有关架构方法的知识，而不得不在项目开发过程中艰难地逐渐学习。显而易见，因为项目架构师们被迫一边自学一边探索，许多项目无法表达出所需的信息。

我们需要掌握有关架构的知识，并使它应用于我们的产业。本书为满足这个需求而跨出了重要一步。

架构工具的迫切之需

世上计算机化程度最低的组织机构可能是 IT 应用系统开发机构。不过，等一下！它们不是在每张工作台上都有昂贵的 PC 和网络连接，并且装配着昂贵的软件开发工具吗？当然，不错。然而它们中相当一部分仍然停留在棉纺产业的工业化水平上。犹如一伙机械工程师被要求用他们的本行工具——焊枪、车床、千斤顶——来制造一种新型汽车。设计方案交予了他们，详细到每一个螺母和螺钉，但是没有针对“这类汽车”架构方法的产品线(或者基础设施)缝合生产过程。他们 80% 的精力用于建造这些产品线，只有 20% 的精力用于生产所需数量的汽车。当他们完成时，早已超过了预算经费和期限。他们的产品线也该被丢

弃了，因为这是为某类特定型号汽车而修建的。

同样，应用系统开发人员也可能具备良好的工具和精深的技巧，但是没有架构方法来教授、规约、定义开发人员的努力。每个项目必须定义它自己的企业架构，并建造自己的基础设施、“粘合”代码、过程定制等(产品线)。当前的工具支持特定的技能与技术，可在多个架构方法间通用。但是，我们没有能够支持特定的架构方法的工具——称为“架构支持工具”。也许这就是我们的开发过程被割裂的缘故：一个可用的过程针对一个架构方法。然而许多通用过程要求额外的缝合和定制。你最近什么时候看到过市面上某个通用客户关系管理(CRM)过程是可以用来解决你的 CRM 的过程需求的？为了提高效率，过程必须是有针对性的——一直到底层步骤。它们必须以制造出你的所需为目的。缺乏此类定位就是当前许多无目标(并且刻意如此)的超重过程被广泛拒用的原因。

我们需要支持工具来支持企业系统所要求的架构方法，本书描述了许多对此类工具的需求。

对适当组织的紧要之需

设想一个 IT 部门拥有一批经验丰富的企业系统架构师、能干而积极的开发人员，以及出色的工具和过程，包括企业架构支持工具。这样就一切齐备了吗？可惜并不是。企业架构师也需要一个合适的人员组织，架构师在其中能如鱼得水地发挥效用。衡量架构师的效用的标准是应用系统开发项目工作的简化程度。

许多 IT 部门在项目(或产品)基础上进行组织。除了一些基本的通用基础设施，比如硬件、操作系统和数据库管理系统(DBMS)，每个项目都要自己决定其余部分。我见过许多创建者以这种方式安排人员和配备设施，但在可能是最困难的部分失败了：人员组织出现了问题。一个解决这个问题的方法是把组织分为两部分。一部分提供开发和运行时基础设施来建立和使用企业应用系统，另一部分则制造企业应用系统。后者尽可能在技术、开发和企业构架方法等诸多方面严格地促进前者。当前业界所采用的组织形式完全不是这样的。

无论组织最后怎么设定，重点在于组织是来支持某个特定的活动的。如果组织不能直接支持和启动这种活动，那么就失败在望了。为了让应用系统开发项目成功运作而采用企业架构，需要妥善考虑人员组织。

本书将介绍有关企业架构的组织形式。

本书的重要性

许多企业系统具有相同的架构方法，这已然成为现今企业架构领导者们间一个鼓舞人心的公论——虽然有些人使用不同的方式来表达，但相似的是都陈述了独立于特定应用领域的种种技术、模式和设计，用来有效地制造出反应灵敏的、可伸缩的、灵活的、标准化

的企业应用系统。本书的重要性在于从多方面概括了这种共性。本书从数据基础要素(即计算机关机后永久保存的部分)到运行时软件设计,到按关注面进行的架构划分,到可伸缩模式,乃至总被忽略的用户界面,全方位地说明了一个成功企业系统所需的知识,并把许多标准综合起来。本书的价值还在于:它不仅讨论了需要建立些什么,而且谈论了如何去建立从工具和模型,到开发过程和方法学,到产品线方法,到敏捷开发,也包括人员组织的重要问题。

此外,本书的闪光点是作者们源自多年经验的、无懈可击的、踏实实践的工作经验和扎实技能。本书包括许多知识点——或者说介绍了许多知识点——采用易读、中肯、不教条的方式讲述,这正是一个成长中的企业架构师所需要的。它是一本基础性著作,可以作为企业架构研究生课程的理想教材。实际上,我预期它能成为在我们这个神奇而生气蓬勃的产业中,企业系统知识领域的重要组成部分。

Oliver Sims
Mortimer West End

序

在如今的商业环境中，竞争的规则已经改变了。当一个新业务问题出现而需要加强已有系统或实现一个系统时，开发组织必须避免每次都重新开发一些小部件。企业架构就是这样一种实践应用，让开发组织建立自己所需的生存环境，以适应现有的和未来的业务。优秀的企业架构支持系统保持敏捷的反应能力，可为投资提供持续增长的回馈，为将来的技术决策创造框架。

企业架构明确了组织所需的主要组件，以及组件如何协同工作于组织的神经系统中，以便实现预定的业务目标。这些组件包括人员、业务过程、技术、财务信息和其他一些资源。如果对于组件和组件间关系的决策未经协调，那么枝端末节的工作和资源都可能重复，这样必将出现性能和管理上的麻烦，导致系统敏捷性(Agility)的缺失。正确实现企业架构能保证选择适合的组件，能规范每个组件流畅地共同工作，提高组织的效率。

建立企业架构是今日信息技术中最富有挑战性的任务之一。此任务的许多方面都与技术有关，但更多方面与交互相关。执行此任务的许多人负有重大的责任，但并不拥有权力或者控制权。企业架构作为一个研究领域，让人们注意到有趣而重要的问题，在合作的阶梯上进行探索，给企业带来巨大变化，同时保持技术上的能动性。

企业架构所需的技能通常无法在大学课程中教授，也难以在工作中获得。即便是优秀的架构师也很少能不经严峻的挑战就被人认可。成功的架构师必须克服来自任何方面的对架构策略的反感，在各方涉众(stakeholder)的眼里成为一个架构的福音传道者。无论如何，架构师不应当仅仅起到简单的衔接作用。任何涉及到企业架构实现的活动都需要真正理解企业架构。那些非常成功的架构后面必定站着一个能够讲述架构选择动机和理由的架构师。亦即，成功的架构的产生必定有一个架构师领导着架构队伍、开发队伍、企业技术方向，有时甚至是组织机构本身。本书的主旨在于成为那些希望胜任架构师角色的人们的指南和可信任的指导者。

为何使用企业架构

一位明智的企业架构师曾经为一个备受尊崇的组织(此处有意隐去其名称)工作。他热情洋溢地和高级主管谈论该组织应当如何发展其信息系统。他的建议一直被置若罔闻，直到他问了主管三个非常简单的问题：

- 当我们还不知道全部的客户数据在何处时，如何建立可行的客户关系管理策略？
- 我们的技术经费是如何均衡分配，来达成业务上的战略目标的？

- 我们的组织有一个运作过程模型吗？

你可以猜到，这些问题的答案尚缺。许多信息技术的主管对如何使用架构来产生最大的财富一无所知。可悲的是，许多主管已经失去对他们所主舵的航船的控制，以至于机构臃肿，决策来自臆度而非基于原则的良好判断和经验。

在许多组织中，年度预算周期始于毫无头绪的、支离破碎的问题，诸如“我们有多少服务器？”或“让我们采用新型 Intel 3333GHz 的台式机。”作为关于企业架构的讨论，这些话无一恰当。一个好的企业架构应阐明如下问题：

- 组织的业务和信息需求。
- 投资回报(ROI)和整体拥有成本(TCO)间协作关系的充分发挥。
- 支持从当前状态移植的能力。
- 支持方便地移植到企业所希求的未来状态的能力。
- 支持节省开销，改善可操作服务，增加回馈等业务目标的各种途径。

许多组织会宣称有一个创建架构的架构师，还有一些组织会企图从五大咨询公司购买一个号称包装良好的架构，但对他们所购得的东西一无所知。如果你的架构师无法阐述如下原则，可能会弊大于利：

- 数据和逻辑分离(业务操作)。
- 出于模块化的原则，确保在企业架构中的每个组件只执行一或两个单独任务。
- 功能被划分为不同类别的可自行满足的任务。
- 架构是可自说明的，如果它不是，则必然有些问题。
- 一个组织制造的所有数据和工件(artifacts)由同一组织管理和维护。
- 架构在技术上是灵活的，能够在合理时间内以合理开销实现。
- 架构从业务需求到技术实现都可被追踪。
- 逻辑架构和物理架构分离。

企业架构模型如图 P-1 所示。

建立企业架构

术语“架构师”(architect)来自建筑业。修建公司的神经系统可直接类比于修建房屋。也许有人会说：“我对如何建造 5000 万美元的摩天大厦一无所知，也不知道如何建造 5 万美元的住宅。”其实这两个任务的答案很简单：他们需要创建设计蓝图来实现。

设计蓝图说明了如何构筑一所房屋。它们提供了多个视图，这些视图表达着各自层次的细节。房屋的其中一个视图会包括所有的电路，它对电工是非常有用的。同样地，管道工需要关于管道和排水系统的设计图。企业架构为开发组织内的所有人员展示设计图，以便让他们知道如何建立一个反应敏捷(agile)的企业。当企业架构蓝图交付到熟练的专业人员手中时，它可为设计的实现提供充分的细节，正如建筑设计图所做的那样。

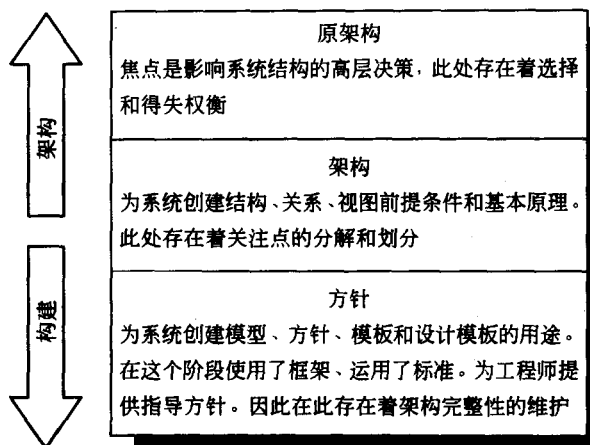


图 P-1 企业架构模型

我们并非暗示构筑摩天大厦和构筑一所住宅一样容易。10年前，Garlan 提到，“随着软件的规模和复杂度增加，计算算法和数据结构以外的设计问题就会出现：设计和制定系统整体结构将成为新的一类问题……这是软件架构层次的设计。” [Garlan 1992]

在此上下文环境中，企业架构寻求智能化管理的解决方案。大项目的架构和其复杂度来自大范围、大规模、资料记录的草创新建、当前采用的技术和其他种种因素。在这种状况下，企业架构的目的是提供一种方式来隐藏不必要的细节，简化难以理解的概念，把问题分割(decomposition)为宜于理解的组件。架构的组成如图 P-2 所示。

组织中出现的复杂度也来自于建筑系统所采用的过程、建造系统的人员数量、必要的外部资源引进、团队的地理分布以及组织基金模型。在理想情况下，通过对关注面进行更清晰的划分和强化交流，企业架构能辅助(但不能消除)对过程的管理，使过程间的彼此依赖性更易于管理。

建造企业架构是困难的，但如果坚持架构原则，也并非不可能。架构蓝图定义了需要建造些什么，而不是如何或者为何去建造。项目计划说明了“如何建造”，项目计划的设立最好留给研究过项目管理的人去做。“为何建造”的问题最好留给业务的服务客户，因为所有的架构都应该是业务驱动的。

一句箴言：良好的架构提供敏捷性但不提供制约。高度成功的架构将去除某些决策，这些决策来自一贯不负责任的冲动决断，或是根据不知情者的观点作出的。一个好架构师会看到让他们头痛的东西，但镇痛药就是明白他们的辛勤工作和坚持能减少潜在的问题。帮助他人看到企业架构的价值是至关重要的，因为即使决策只产生部分的成果，也能给企业带来福祉。

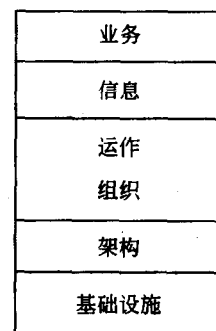


图 P-2 架构组成

关于本书

本书的产生来自于它的其中两个作者一次午餐谈话。他们追思着为何计算机领域是惟一个不需要任何内行经验就可成为经理、领导，甚至副总裁的领域。在这次午餐会谈中，我们把计算机领域和其他一些专业领域做了比较，比如会计行业、司法与执法行业以及医药业。

试想你所居住的镇需要招聘一位警长，如果他从来就没当过警官，你能信任他吗？更有甚者，如果这位准警长不知道如何使用枪支，对刑侦程序也一无所知呢？以此类推，想象一下一位前麦当劳快餐店经理申请成为会计或律师事务所的合伙经营人，仅仅因为他当过领导，但是他仍然可以固守对自己专业的现有经验，讨论案例或制衡他人。这位准警长作为领导，根据他的相关经验，最可能记得如何开停车罚单，或诸如此类街区警察的任务。

对这个问题刨根问底需要花点时间。计算机领域不像其他领域，你可以有一个架构师的头衔而不必真正了解你的工作。信息技术领域处于持续不断的发展变化中。基于这个原因，它要求深思熟虑的产业领导者花费大量时间阅读许多书籍。即便对最勤勉的人而言，求知和启迪的目标也永难完全实现。

许多业务都面临着技术迅疾变化的挑战。技能有限的员工、更有限的经费预算，以及对失败极少的容许度。在这个新秩序下获得成功的惟一途径是采用新策略。今天的业内人士不再有机会以失败为成功之母了。

本书介绍了几个富有启迪性的策略，可作为引领、教诲和向导，助你管理千头万绪。它将挑战保守和顽固的思维。本书由几位最睿智的信息技术思想领袖撰写，旨在帮助你在商战中运用所学来创造一个反应敏捷的企业。

祝您成功！

本书内容

本书的目的在于以一种易读和实用的方式，分享业界思想领袖所积累的知识。本书包括许多新锐实例，表现企业架构如何应用于现有业务和技术问题，有助于在解决当前问题的同时关注如何贯彻和考虑企业架构。本书涵盖了如下论题：

- 系统架构
- 软件架构
- 面向服务的架构
- 产品线实践
- 方法论
- 企业统一过程

- 敏捷架构
- 敏捷建模
- 表示层架构
- 用户经验和可用性
- 数据架构
- 思想领袖

图 P-3 列出了 McGovern/Steven 架构敏捷性所包含的内容。

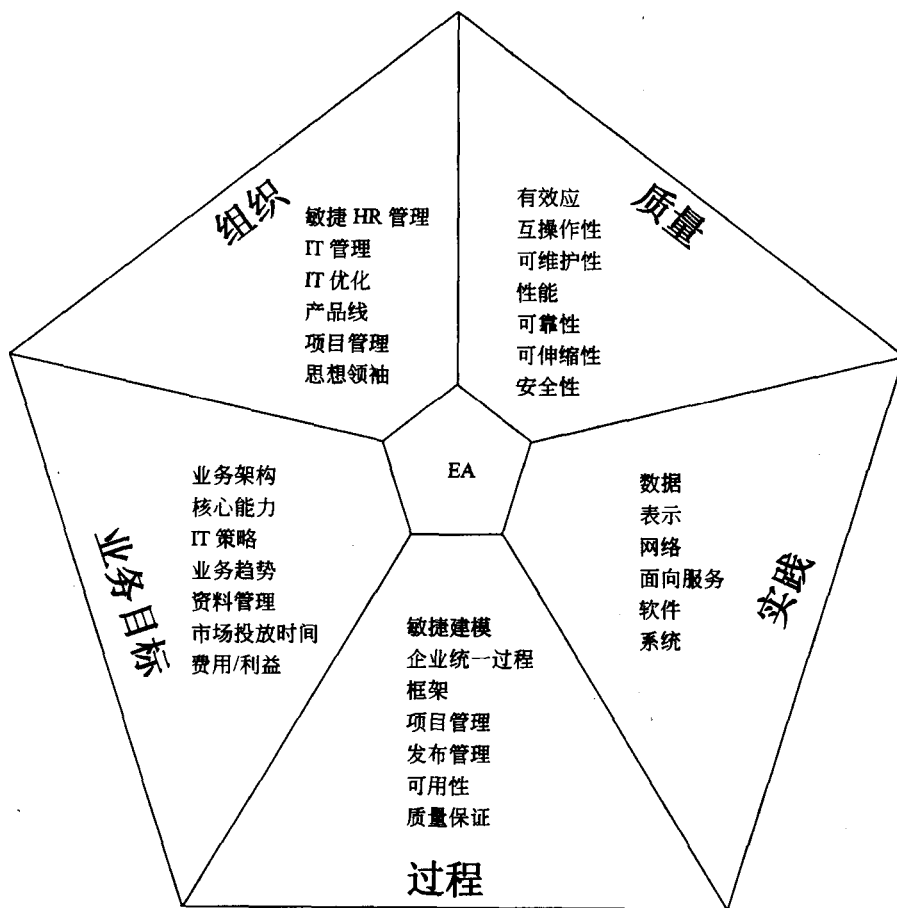


图 P-3 McGovern/Steven 的架构敏捷性五角形

作者和读者可能对如何最有效地阐述一个给定论题各有偏好。本书包含的实例不是对这些主题极为详尽的描述，现在有不少书更切合这个定位。我们会重点讨论提供额外增值经验的内容。

本书的作者队伍也相信这不是一本可以漫不经心一目十行浏览的书，本书包含的内容

要求深入的研究甚至贯彻执行，方可完全被了解。

读者对象

本书适合一切充满激情和经过学术训练的 IT 专业人员，本书可用来探究成功的项目实现所需要的技术。本书的理想读者应是财富 1000 强企业或者服务于这些企业的咨询公司所雇佣的项目经理、资深开发人员、软件架构师或企业架构师。本书也适合 IT 技术管理链中的执行主管，如首席信息主管(CIO)、首席技术主管(CTO)等。

因为架构师不是一个基于执业资历而委任的头衔，所以作者有意避免给出此职业的任何有关年资的参照数据。我们假定读者基本了解新兴的技术及其在解决当前问题中的作用。如果读者有兴趣在某些特定领域深入探究，那么可参照附录 E 中提供的参考文献。

弃权声明

只要你不冒称是作者，就可以随意使用本书所含的建议、图表和推荐意见。出版者、作者及其各自的雇主不提供任何形式的授权，也不担保此书在任何特定目的上的使用。

本书百分之百正确——当然不是这样！你要能相信这句话哪怕一秒钟，我就把我家后院里那个吊桥卖给你。作者队伍和编辑们竭力工作，以求呈现给你一本有关企业架构的、易于理解的正确指南。如果你在本书中发现任何错误，请用 eaauthors@architectbook.com 与我们联系。

本书使用一个虚构的汽车制造企业来帮助说明实例。其中所有的公司、组织、产品、域名、电子邮件地址、人员、地点和事件描述都是虚构的。请勿有意且不应有意将它们和任何真实的公司、组织、产品、域名、电子邮件地址、人员、地点和事件关联起来。

小结

开发一个企业架构是困难的任务。要求从重叠和矛盾的需求中加以提炼，从这些需求中创建良好的抽象概念，形成一个有效、经济的实现过程，以及巧妙地解决孤立编码和抽象的棘手问题。更进一步说，企业架构师们必须关注所有这些工作直到最后成功，所有这些必须在尽可能最低的费用和时间开销内完成。真能实现如此的惟一方法就是保持敏捷性。

在一个敏捷的方案里，架构师的任务设定不包括建模和概念设计。他们必然极少为项目经理提供技术指导服务。这可以类比于建筑领域中建造工程师们和建筑设计师之间的关系。

本书陈述的许多思想会设定一个敏捷方法，来说明所谈到的问题的解决。对软件开发敏捷方法的理解，将会帮助读者在实现实用企业架构中更为成功。