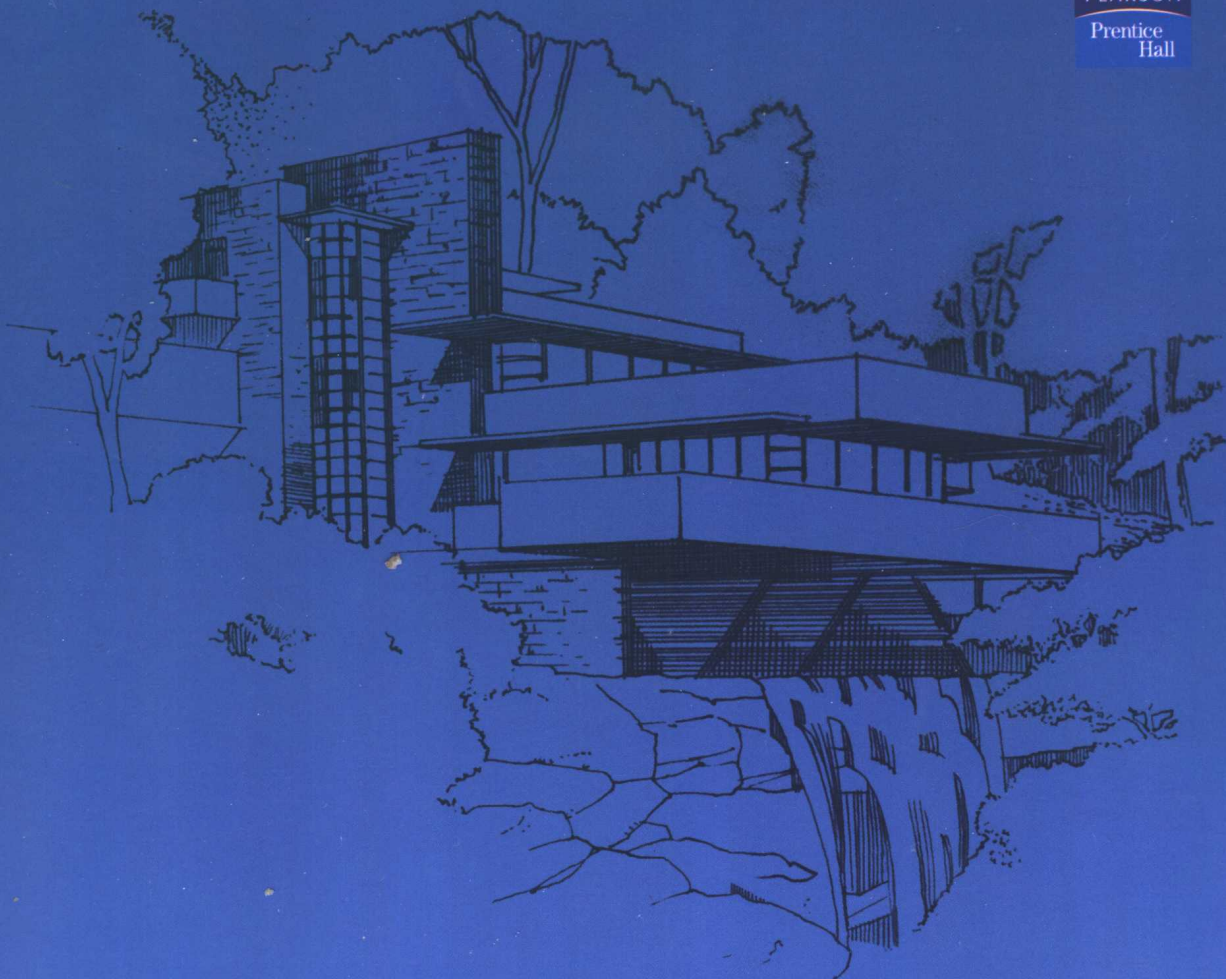




软件工程实践丛书

PEARSON
Prentice
Hall



需求分析

Requirements Analysis
From Business Views to Architecture

(美) David C. Hay 著

孙学涛 赵凯 朱卫东 译



清华大学出版社

软件工程实践丛书

需求分析

(美) David C. Hay 著
孙学涛 赵凯 朱卫东 译

清华大学出版社

北京

内 容 简 介

本书首先明确了需求分析的目的及其重要性。然后作者通过介绍、分析现有的多种技术,来解释如何进行有效的需求分析。书中为读者掌握从商业角度到软件构架的过程提供了有效指导。

本书可作为软件学院及大学计算机等专业相关课程的教材,也可以作为软件公司各级管理和开发人员参考资料。

Simplified Chinese edition copyright © 2004 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

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

Original English language title: *Requirements Analysis : From Business Views to Architecture*,
1st Edition by David C . Hay, Copyright © 2003

EISBN: 0-13-028228-6

All Rights Reserved.

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

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 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: 01-2003-0547

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

需求分析/ (美)赫(Hay, D. C.) 著; 孙学涛, 赵凯, 朱卫东译. —北京: 清华大学出版社, 2004.

(软件工程实践丛书)

书名原文: *Requirements Analysis : From Business Views to Architecture*

ISBN 7-302-08404-1

I. 需… II. ①赫… ②孙… ③赵… ④朱… III. 软件开发—系统分析 IV. TP311.52

中国版本图书馆CIP数据核字(2004)第026908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

文稿编辑: 李强

封面设计: 立日新设计公司

印 刷 者: 北京市昌平环球印刷厂

装 订 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 24.5 字数: 593 千字

版 次: 2004年5月第1版 2004年5月第1次印刷

书 号: ISBN 7-302-08404-1/TP · 6043

印 数: 1 ~ 4500

定 价: 39.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

译者序

需求分析就是对系统开发工作提出明确要求的过程，是软件开发的一个重要环节，其结果将成为后续的系统开发工作的基础。软件开发人员都清楚需求分析的重要意义，或至少知道有这样一个环节。但在实际开发时，经常由于种种原因而在需求分析这一环节偷工减料，其结果是在后续阶段付出更高的代价。

需求分析又是一项复杂的工作，其实践经常缺乏有效的理论指导，将分析与设计混为一谈者也不鲜见。本书不是简单地阐述需求分析的某种（些）技巧，而是以更广的视角，全面而清晰地阐述了应如何进行需求分析。在系统开发生命周期和软件构架框架两个概念的基础上，借鉴前人的研究成果，给出了构架框架，指出需求分析过程中所运用的任何技巧（无论本书是否讲到之）都可填写到此框架矩阵中。全书内容按照这一框架而展开，对需求分析过程中可运用的各种方法和技巧做了全面介绍和精辟分析。这一框架既可使读者清楚地认识到系统开发是由具有不同观点的若干类人员共同完成的这一基本事实，又可帮助读者加深对每一种开发技巧的认识，为需求分析实践提供了明确的指导。

本书作者 David C. Hay 在信息技术领域经验丰富。自 20 世纪 80 年代中期以来，一直潜心从事用于战略信息规划和需求规划的数据模型的研发，并出版有关数据模型方面的专著。他的公司——Essential Strategies 公司——致力于帮助客户确定信息构架、搞清需求及规划如何采用更先进的系统。

本书第 1~5 章由孙学涛翻译，第 6~8 章由赵凯翻译，附录 A、B、C、D 由朱卫东翻译。孙学涛还翻译了前言、绪论等内容，并负责全书的统稿工作。参加翻译的还有王中尚、张力等。计算机文献的翻译在语言和计算机专业两个方面对译者都有较高的要求，虽然我们付出了巨大的努力，但书中可能仍有不妥之处，欢迎读者批评、指正。

译者

2004.1.30

序

在我儿子 5 岁时，他与那个年龄的大多数孩子一样，对周围的世界充满了好奇。我在读 Dave 的这本书时，他天真无邪的样子又浮现在了我脑海中。当时我们一家人正在划船，儿子问船为什么能够浮在水上，而不会沉下去。

为了说明当时的情况，这里有必要交待一下：我丈夫是土木工程师，我是搞数学的。所以，上面这个问题正是我们乐于和孩子们探讨的。实际上，当时我们对儿子的这个问题确实表现出了极大的热情：很精辟地介绍了阿基米德定律，并用该定律的输入、输出参数做了全面的讲解。

那么，熟悉 5 岁儿童的读者已经知道这样做的结果了。当时儿子立即变得不耐烦起来，他对阿基米德根本不感兴趣。他仅仅是想知道：是船的原因使之能够浮在水上，还是水的原因把船托了起来？

这确实是让我们做父母的感到困惑的典型情况之一：当时我和丈夫哑口无言。这是什么问题？应怎样回答？

当然，John Zachman 的框架为解决我们的这种困惑提供了思路。他的框架告诉我们：对同一现象，我们可以用多达 6 种角度去研究。业务专家和信息构架师的看法不一样，信息构架师和系统设计师的观点也互不相同。更重要的是，每种看法都是合情合理的。

通常，对同一事物（如上面的小船），观察者总是用他感到最直观的角度去观察它。显然，可怜的儿子当时最希望采用一种直接、具体的视角，弄清所关注的对象受到控制的原因。而我和丈夫当时则更喜欢用所涉及的科学基本定律（或业务策略和父母的道理！）去解释它。

但这两种（及其他一些）看法对这艘小船来说都是合理的——即使当时我们对此还不十分清楚。

在过去的数十年间，因看法的不同而导致的混乱令信息行业的从业人员备受磨难。今天，有了 Zachman 框架，我们对这一问题认识得更清楚了。换言之，现在我们知道：如果两个人讨论同一个待开发的系统，但却相互不理解，很可能是因为他们各自考虑的是这一框架中的不同内容。未必哪一个人就是错的或考虑不全面，他们只是从不同角度看待同一问题而已。

但 Zachman 框架则是一种总的视图，其中包括各部分结果在总蓝图上（或者在如 Dave 在本书绪论中所说的乐谱中）如何关联及在现实世界中如何关联。

要创建这样的蓝图是相当困难的。Zachman 框架清楚地表明了这一挑战的广度和深度。但作为具体实践者，我们该如何按照这一框架展开工作呢？直到现在，读者还是靠自己的摸索。但是，Dave 的这本书为我们提供了极好的指南。而且，Dave 还在书中对这一框架的各个部分所涉及的各种技术做了全面介绍，从而明确了如何把设想转换成构架。

该书的重要价值体现在 3 个方面。首先，它既适用于新手，又适用于有一定经验的人。因为本质上，此书是对历史的回顾，是一本教程。其次，它对任何读者都有一定价值——无论读者对某个特定系统开发持什么态度。这是因为该书把读者熟悉或不熟悉的学科中所采用的技术很好地结合了起来。其中既有已经验证过的结构化系统分析技术，又有信息工程技术、面向对象技术和最新的商业规则方法。该书作者希望此书成为定义从设想到构架的路线图的权威之作。该书重点突出，条理清楚（这是 Dave 的写作风格）。我们可能要感谢 Dave 对这一领域的见解，也感谢他以这样正式而又有趣味的形式与我们共享这些成果。

第三，最重要的是，该书提升了（很有必要！）需求分析工作的重要性。为什么需求分析这么重要呢？

Dave 指出：在软件开发行业，我们经常是在没有明确要求的情况下工作的，这很让人烦恼。我们就是即席创作者的典型代表。Knowledge Partners 公司的 Bob Giordano 指出，这并非什么新情况。他说：“我们一直用重复的方法来开发。在 20 世纪 70 年代和 80 年代，由于独立系统总是把所有内容耦合到一起，所以每一次这样的重复要历经数年，耗资上百万。”

今天，这种重复现象更明显、更普遍，或许随着近年来采用重复开发方法的日益增多和范围的扩大，更加剧了这一现象。毫无疑问，这些趋势对系统开发的某些方面——对那些不确定的或随意性的部分——是有益的。但是否暗含有以牺牲分析的基本要素为代价的情况呢？或许是有的。

Dave 指出，我们是为了最终结果的简单而进行分析。需求分析的目的之一就是为了防止开发出实际上没有必要的复杂系统。分析还可以帮助我们搞清将要发生改变的成分。当今世界，变化的速度越来越快，存在着各种不确定性。当我们开发的系统投入使用时，世界的变化不会终止。因此，我们开发的系统应具有适应变化的能力，使变化永远成为这场游戏的主题——并且不必付出过高的代价。否则，我们的系统可能会有效地使用到某个时间，然后就开始起消极作用。

我们正在用信息技术确定商业变化的速度和方向。我们应该遵循某种规定，使充满不确定性的未来能够有条不紊地展现在我们面前。在创建这种规定时，以本书为指南吧。

时间过得真快。儿子提出船为什么浮在水上这一问题已经是许多年前的事情了。可以想见，儿子可能早就忘了曾经提出过这一问题。已经长大的他像所希望的那样，通过阿基米德定律对这艘船的行为有了更深刻的认识。

到底应该怎样回答儿子当年提出的这一问题？我仍然不知道。因为对他当时的看法，

我并不完全清楚，而只知道其中的某些部分。

但是，如果想要搞清这一问题的答案，我可以到上面所说的蓝图中去找。

这样找到的答案会是很完美的。同时，由于其他人的工作，这种答案似乎也是简单的。正如读者将看到的那样，在这种蓝图上的投资是非常有意义的——因为它代表了分析上所做的深入思考。这种思考一方面考虑到了系统的简单性，另一方面也考虑到了适应变化的能力。有了这两方面（简单性和灵活性）的考虑，这种蓝图（或乐谱）就能够经受得住时间的考验。如果跳过了需求分析，就无法享有这些成果。Dave 将帮您搞清楚这一切。

Barbara von Halle

前 言

编写此书的源起：面向对象的分析

近年来，面向对象的编程极大地减少了开发系统所需的时间和精力。作为一种来自实时系统的技术，面向对象技术已经成功地应用到了基于视窗的交互式界面及其所支持的系统。面向对象技术的优点之一是模块化程度高，这使得我们可以在无需对整个系统做大的改动的情况下，方便地构建和修改系统的某些部分。而且，模块还可以重用。

同时，面向对象的思想已经影响到了需求分析这一领域。市场上已经出现了许多本关于“面向对象分析”的著作；而且，作为对这一技术的支持，UML 也走上了历史的舞台。

但与这一状况相伴的，有 3 个问题。首先，像所有其他程序员一样，采用面向对象技术的程序员，一般更多关注的是编写程序的技术，对分析某个企业的本质并无多大兴趣。进行这种分析需要其他一些技巧，而这些程序员不大可能具备这些技巧。似乎有人认为面向对象的设计师就是自然系统的分析师，而实际上未必如此。

其次，有些作者认为需求分析是面向对象技术所特有的现象，而并没有注意到来自信息工程或其他方面的一些思想的作用，尽管它们与面向对象领域无关。这种想法有很多问题，其中一点就是简单地忽略了这些数十年来对面向对象领域一直都具有重要作用和价值的学科。

例如，1991 年，James Rumbaugh、Michael Blaha、William Premerlani、Frederick Eddy 和 William Lorensen 共同出版了 *Object-Oriented Modeling and Design* 一书。此书中给出了一种对象建模表示法，同时还给出了一种称为“对象建模技术”(Object Modeling Technique, OMT) 的方法。这种表示法中包括若干符号，其中有些表示的是与 10 年前 Clive Finkelstein 和 James Martin 在某信息工程文献中相同的概念。OMT 考察的是对象类（实体类型）、关联（关系）和属性。

这种方法虽然号称与“传统软件开发方法”[Rumbaugh 等，1991，第 146 页] 有着很大不同，但实际上与信息工程方法很相似。像信息工程方法一样，这种方法也是基于“把开发工作转移到分析”、“在考虑功能之前强调数据结构”和“无缝开发过程”等原则的。而这些原则就是 Finkelstein 先生和 Martin 先生早已申明的原则。这两种方法之间的一个比较明显的差异是 OMT 方法具有更多的面向交互的特色。

在 *Object-Oriented Modeling and Design* 一书的第 12 章，提到 Peter Chen 是实体 / 关系建模方法的创建者，并称对象建模技术直接来自于实体 / 关系建模方法 [Rumbaugh 等，1991，第 271 页]。但此书的其他部分并未体现出这一点。

现在的需求分析实际上综合了 30 多年来许多人的贡献。需求分析在系统开发生命周期中的作用比以往更为重要——即使是采用了面向对象技术。与某些人的说法不同的是，需求分析并没有随着这些技术的出现而发生根本变化。

面向对象的思想确实对需求分析有所贡献，但这种贡献要比某些人的说法小得多。需求分析就是要搞清商业上对信息技术的需求，而不是为了确定应用哪种技术来解决这些需求。正确的需求分析结果应能够指引设计师使用任何技术。

第三个问题则来自那些坚持认为需求分析中应包括“面向对象特色”的作者。“控制对象”和“接口对象”都是来自面向对象设计的，但经常被（不合适地）当作需求分析的一部分。

实际上，确定需求的过程与应用某些技术来满足需求的过程有着根本区别。在不考虑（除了泛泛的了解之外）采用什么实现技术的情况下确定出需求应该是可能的。对同样的一组需求，可能既可以用面向对象技术来实现，也可以采用 Oracle 公司的关系数据库工具来实现，还可以用某组 COBOL 程序来实现。

关于本书

本书没有从某一特定实现技术的角度来看待需求分析，而是把需求分析看作是构架层次上的一个过程。具体来说，需求分析就是要回答以下的问题：我们怎样才能搞清企业的体系结构，从而使所要开发的系统能够真正支持这一构架？为此，本书尽可能多地介绍了整个系统开发历史上出现的最优秀的技术和方法（包括来自面向对象领域的某些技术和方法），并将讨论在开发面向对象的系统或其他系统时应用这些方法和技术是否合适。

Merriam Webster 把“构架”定义为“某种统一或一致的形式或结构” [Merriam Webster, 2001]。本书为需求分析而描述某构架时，就是在描述整个需求过程的结构。本书受到了 John Zachman 所做研究的影响，他用一种矩阵来描述系统开发的构架。这一矩阵的行对应于开发过程中各类人员的观点，列则对应于将从各个角度考察的事物。本书所讲述的各种技术对应于这一矩阵中的各个单元格。

本书假定需求分析就是把某企业的企业所有者视图转换成关于该企业的一个全面的构架视图。本书开头几章是一般介绍，之后对从这两种角度看的每一方面（什么、怎样、何处、谁、何时、为什么）各用单独一章讨论。

虽然本书主要关注的是上述两行，但对“范围一行”也必须予以重视，它把我们所做的全部工作都考虑在内。另外，为便于理解（但仍坚持上述观点），也会偶尔提及对采用

某一技术的设计师来说应该受到的启示。

尽管从某种意义上说，本书将涉及某些已有人研究过的领域，但本书独具特色。这是因为本书对分析项目可能给出的各种结果都做了介绍。这种全面性是绝大多数同类图书所不具备的。本书不仅讨论了对数据和处理的分析，而且还阐述了对数据网络、人员与组织、事件、动机等方面的分析。

本书并不是像推理小说那样可以匆匆看完的作品。绪论、第1章和第2章对这一领域做了全面介绍，并指出了应预先阅读的文献。其余6章则既是路线图，又是参考手册。所以说它们可以起到路线图的作用，是因为它们对构架框架36个单元格中的12个做了一一介绍，并阐明了它们之间的相互联系；说它们是参考手册则是因为对所讲述的每个单元格，都介绍了可采用的多种技术。如果读者曾听说过某个技术，想对其深入地了解并弄清这种技术在全局中所起的作用，应该能够在这些章节中得到满意的答复。

应注意的是，Zachman先生的伟大发现就是要提供一种能把我们所知的内容组织起来的框架。在我们填写这一矩阵的内容时，很快就会发现对各个单元格的熟悉程度是不一样的。在阅读此书的过程中，读者将会发现这些章节对相应内容的阐述并不同样完美或同样全面。实际上，这些章节所占的篇幅也是不一样的。而且，每个章节在写作风格上也都有所不同，这取决所掌握的关于相应列的信息。

例如，我们在数据建模方面有着很多经验，在位置或人员与组织建模上的经验则相对较少。Codd先生为我们奠定了数据建模的数学基础，而对其他列来说则没有类似的数学基础。对于其他列，我们是在努力建立某种规律，但难度很大。

有位同事曾经说过，希望有一本书能够“帮助像他这样没有时间来学习并非是新内容的各方面知识的人”。我想本书就算是一本吧。在本书中，将阐述一些已有的内容，同时也将指明未来发展的方向。

能有机会从事某个正在发展中的行业是我们这个时代的乐趣，同时也是一种烦恼。如果幸运的话，本书的后续版本中将会添加更多的内容。每位读者都应努力为该领域知识的扩展贡献力量，这算是本书留的作业吧。

重要说明

本书并不是要讨论哪种技术更好，而是试图介绍各种技术，让读者根据自己的情况去选择。迄今为止，在我们这个行业当中，用于数据建模和行为建模的技术最多。但本书最重要的章节并不是关于数据建模或行为建模的。第5章是关于人员与组织的^①，其中讲到他们对企业的成功与否所起的作用，以及保证这种成功的通信信道的重要性。这方面的内

^① 译者注：原文的“6”应为“5”。

容尚未很好地或很系统地讨论过。

因此，第 5 章追溯到 20 年前的控制论领域，并运用该领域的一些成果来讨论构成维系企业生命所必需的通信信道。具体地说，该章描述了用“放大器”或“衰减器”（过滤器）来管理每个企业所面临的“多样数”（复杂性）的急剧增加。毕竟，管理多样性是每个人实际上所做的工作。在这一章的中间部分，给出了对需求分析工作的真正目的的表述：

需求分析就是通过考察某个组织机构而确定出要构建的最有效的放大器和衰减器。需要什么？现有系统是怎样表现得无效或达不到预期目标的？考虑到企业的目的和组织结构，所要开发的系统应该是什么样子的？

放大器和过滤器？这是什么意思？——读者可能会这样问。要想搞清楚，就读读第 5 章吧^①。

致 谢

本书更像是对他人成果的概要介绍，而并非完全原创。书中所列出的人员当中，有些为本书的撰写提供了很大的帮助。其中，首先有最早提出构架框架的 John Zachman。他非常乐于同我讨论某些名词的含义，对此我深表谢意。另外还有创建对象角色建模技术的 Terry Halpin、IDEF0 方法的权威 Clarence Feldmann、实体生命历史的原创者之一 John Hall、商业规则女王 Barbara von Halle 等。然而，本书可能有的任何错误都是本人的责任，与他人无关。

我也要感谢 Edwin Landale。他详细审阅了书稿的全部内容。在很多问题上，我们的看法是一致的。这一点在他发现不足时表现得尤为突出。Mark Gorman 和 Pauline Hannemann 也都审阅了全部书稿，并给我提出了一些很有帮助的建议。Beau Cain 很耐心地检查了语言表达上的问题。

Spencer Roberts 和 Scott Ambler 帮助我解决了面向对象领域的一些问题。在很多方面我们的意见有分歧，但这对强迫我自己保证论述的严格性特别有帮助。^②

多年来，商业规则组一直在把握企业动机和商业规则方面给我以帮助。另外，该组织的 Keri Anderson-Healy、Michael Eulenberg 和 Dennis Strunk 审阅了本书的某些章节，提出了宝贵的建议。

加利福尼亚州巴克思菲尔德市 Aera Energy 公司的员工为我提供了本书第 3 章所给出的项目规划的细节，在此深表感谢。这家公司确实采用的是科学的做法：他们构建了完整的企业构架战略，并在此基础上展开若干具体的需求分析项目。他们对所采用的技术有着

① 译者注：原文的“6”应为“5”。

② “与持相同意见者相处，会感到舒服；与持不同意见者相处，会变得成熟。”——Frank A. Clark。

深刻的理解，在工作中表现出极大的热情。

还有其他一些人士也审阅过本书的不同部分。其中包括：Mike Lynott、Anne Marie Smith、Paul Dorsey、Lwange Yonke、Edwin Landale、Marc Danziger、Cheri Howe、Scott Warren、Bob Schmidt、Diane Blaum、Kathi Bean、Roger Gough、Alan Nitikman 等。

感谢 Prentice Hall 出版公司的 Greg Doench、Donna Cullen-Dolce 和 Bob Lentz 为本书的顺利出版所做的工作。

我母亲——Henrietta Hay——虽然已有 80 多岁的高龄，仍然在为著名的科罗拉多州 *Daily Sentinel* 杂志撰写每周专栏。她住在一个传统的小镇上，而她自己却是个自由主义者。她非常活跃，善于激励自己。在我幼年时期，她向我讲授了在写作中讲究方法和保证条理清晰的重要性。她的这些教导使我受益至今。谢谢您，妈妈！（哦，她多年来为 Mesa 郡公立图书馆工作——本书中多处出现的关于虚构的恐龙郡公立图书馆的例子就是以该图书馆为蓝本的。）

当然，我还要感谢我亲爱的妻子 Jolanta。感谢她的支持，也感谢她为本书所做的编辑工作。还要感谢我们的孩子 Pamela 和 Bob，感谢他们为我收集计算机程序而做出的贡献。

要特别感谢 Bob。他是要求最苛刻的编辑。出于某些原因，他直方不讳地批评父亲的作品——对此我也感到自豪。

David C. Hay
美国德克萨斯州休斯顿市
dch@essentialstrategies.com

目 录

第 1 章 软件构架框架	11
1.1 Zachman 框架.....	11
1.2 构架框架.....	13
1.3 分析过程.....	15
1.4 含义.....	17
第 2 章 项目管理	19
2.1 引言.....	19
2.2 开发阶段概述.....	21
2.3 关于战略.....	24
2.4 关于需求分析.....	25
2.5 工序 1: 确定范围.....	26
2.6 工序 2: 规划工序.....	26
2.7 工序 3: 收集信息.....	27
2.8 工序 4: 对企业进行描述.....	30
2.9 工序 5: 确定对新系统的需求.....	43
2.10 工序 6: 确定现有系统的环境.....	50
2.11 工序 7: 筹划变迁.....	52
2.12 小结.....	56
第 3 章 第 1 列: 数据	57
3.1 对数据的看法.....	58
3.2 数据构架简史.....	59
3.3 高级数据管理——元数据.....	68
3.4 图——数据建模.....	68
3.5 使用实体 / 关系或对象模型.....	80
3.6 规范化.....	84
3.7 数据建模约定.....	102
3.8 实体 / 关系模型的验证.....	119
3.9 需求分析提交的结果——第 1 列.....	122
3.10 数据与其他列.....	124
3.11 结论.....	125

第4章 第2列：行为	125
4.1 从企业所有者视图到构架师视图.....	126
4.2 途径.....	128
4.3 功能层次图.....	129
4.4 依赖关系图.....	131
4.5 数据流图.....	134
4.6 IDEF0.....	147
4.7 UML 行为图.....	156
4.8 交互图.....	158
4.9 用例.....	160
4.10 关于业务处理反向工程.....	160
4.11 功能和处理的详细文档.....	162
4.12 行为分析的隐含的意义.....	166
4.13 需求分析提交的结果——第2列.....	167
4.14 行为与其他列.....	168
第5章 第4列：人员与组织	171
5.1 企业该如何组织（第1行）.....	172
5.2 第2行：企业所有者视图.....	173
5.3 第3行：（人的）系统的本质.....	179
5.4 该模型的含义.....	200
5.5 系统使用.....	202
5.6 需求分析提交的结果——第4列.....	204
5.7 人员、组织和其他列.....	205
第6章 第3列：位置	207
6.1 第2行：地理.....	208
6.2 第3行：网络（和其他列）.....	211
6.3 需求分析提交的结果——第3列.....	213
第7章 第5列：时序	215
7.1 引言.....	215
7.2 第1行：范围.....	216
7.3 第2行：企业所有者视图.....	216
7.4 第3行：构架师视图.....	219
7.5 需求分析提交的结果——第5列.....	238
7.6 时序和其他列.....	238
7.7 结论.....	239

第 8 章 第 6 列：动机	241
8.1 引言	241
8.2 第 1 行：范围	243
8.3 第 2 行：企业所有者视图	244
8.4 第 3 行：构架师视图	250
8.5 需求分析提交的结果——第 6 列	281
8.6 动机与其他列	282
8.7 结论	283
附录 A Zachman 框架	285
附录 B 数据建模技术比较：句法约定	287
B.1 Peter Chen	289
B.2 信息工程	293
B.3 Richard Barker 表示法（Oracle 公司所用版本）	295
B.4 IDEF1X	299
B.5 对象角色建模（ORM）	306
B.6 统一建模语言（UML）	311
B.7 可扩展标记语言（XML）	317
B.8 建议	322
附录 C 商业规则组动机模型	327
附录 D 改进的商业规则组动机模型	329
术语表	331

绪 论

几年前，我观看过休斯顿歌剧院的一次彩排。那次排练的剧目是 Richard Wagner 的歌剧《罗恩格林》。那天，在偌大的休斯顿沃塞姆剧院里，大概共有 5 名观众。当时的情形让我觉得这场歌剧就是专门演给我看的。那次彩排给我留下了极为深刻的印象。

在彩排过程中，先后换过几次布景。在主人公乘着天鹅拉的小船登场（只是一种形象表演——并未真的用天鹅）之前，又一次换布景，而且用了大约 30 分钟。在此期间我开始琢磨：我到底是在看什么呢？除了表演此剧目的十几个主要演员外，还有 75 个唱诗班歌手，伴奏的乐队也有 100 多人。另外还有大约 50 人的各类技术人员也是演出所必不可少的，其中包括舞台工作人员、灯光师、负责播放台词字幕的人等。这还没有计算负责布景设计与制作的人员、化妆师、服装师等。当然，当时在场的还有休斯顿歌剧院的一些管理人员。屈指算来，总共有近 300 人在忙碌着，其目的就是为了演出这一剧目——这是我有生以来看过的最精彩的舞台演出之一。

而在软件开发这个行业中，能使 3 个人这样密切地合作就很不错了。为什么会是这样的呢？

《罗恩格林》这场歌剧的演出能使这么多人合作的根本原因当然在于剧作者 Richard Wagner。大约 150 年前，他编写了这一剧目，并在剧本中做了非常详细的说明。特别值得注意的是，他还为此剧谱写了乐谱和歌词。剧本中对每个演员、每个唱诗班歌手及乐队的每个成员都有明确的要求。当然，布景设计师享有一定的自由。此剧的布景设计师 Adrienne Lobel 采用的是超现实主义画家 René Magritte 的风格。这当然使舞台布景颇具特色。但即使是 Wagner 在剧本中没有做明确要求的舞台工作人员，也要根据布景设计和舞台上的场景来完成各项任务。

而在软件开发行业，我们经常缺少的就是这种明确的要求。

需求分析就是对系统开发工作提出明确要求的**过程**。开发工作的目标是什么？系统由哪些组成部分？应该由什么人来完成什么工作？如果没有明确要求，开发团队的每个成员就会按照自己的特定想法来完成手头的工作。这样做的结果是各个成员所做的工作不能很好地协调，也不能构成一个有机的整体，从而经常导致整个系统无法运行。这样开发出来的软件产品当然不可能存在 150 年。

若干年前，程序员们为了实现某个具体任务而编写软件。当时如果某个程序员知道要完成什么样的任务，他自己就可以编写出一个软件。这种即席创作完全能够满足当时的需要。那时候的编程不像上面所述的歌剧演出，而更像是一场爵士音乐会。但现在我们要开发的软件系统，则是要成为某个组织的基础结构的一部分。如果不清楚该基础结构的实质、

2 需求分析

不清楚所开发的系统将发挥的作用，我们就无法顺利完成这种软件的开发。这就像没有乐曲就不可能演出歌剧一样。

目前，软件开发行业有一种因系统开发受到各种压力，而在需求分析阶段偷工减料的趋势，这十分令人遗憾。开发团队没有能够在开始系统开发前，坐下来好好讨论一下要开发的系统将会是什么样子，应该如何实现。这并非是因为软件开发团队不知道该怎样做这一工作，而是因为在实际开发中经常面临一些相互冲突的需求，迫使软件开发走捷径，跳过需求分析这一步。

这样做的结果，总是不得不在开发的后续阶段付出更高的代价。显然，这样开发出的系统的生命力也不会像前面提到著名的歌剧那样强。

在需求分析阶段偷工减料将导致的一个主要问题是开发出来的系统很复杂，而且这种复杂实际上是不必要的。简单系统的开发要比复杂系统容易得多，但简单系统的设计则要难得多。这是非常重要的一点。

我们必须清楚问题简明的真正含义，而做到这一点其实不容易。

需求分析应由能够关注出资企业的本质及其在信息方面的需求的人员来完成。那些痴迷地认为用某项技术可以解决所发现的任何问题的人不适于做需求分析工作。

例如，我们可以考察如下的这首小诗：

*Un petit d'un petit
S'etonne aux Halles
Un petit d'un petit
Ah! desgrés te fallent
Indolent qui ne sort cesse
Indolent qui ne se mène
Qu'importe un petit d'un petit
Tout Gai de Reguennes.*

——Luis d'Antin van Rooten

Mots d'Heures: Gousse, Rames [Beer, 1979, 第 301 页]

如果读者懂法语，就会发现这首诗是无法理解的。它看似用法语写的，也具有法语的语法特征，但按照法语来理解又是完全错误的，在法语中毫无意义（“在 Halles，少许的少许让它自己感到诧异”？）。另一方面，如果读者自己不懂法语，但有懂法语的朋友，则可以让这位懂法语的朋友大声朗读一下这首诗。如果不懂法语的读者仔细听，就会明白这首诗的含义了^①！

① ……如果还不明白，在本绪论的最后有提示。