



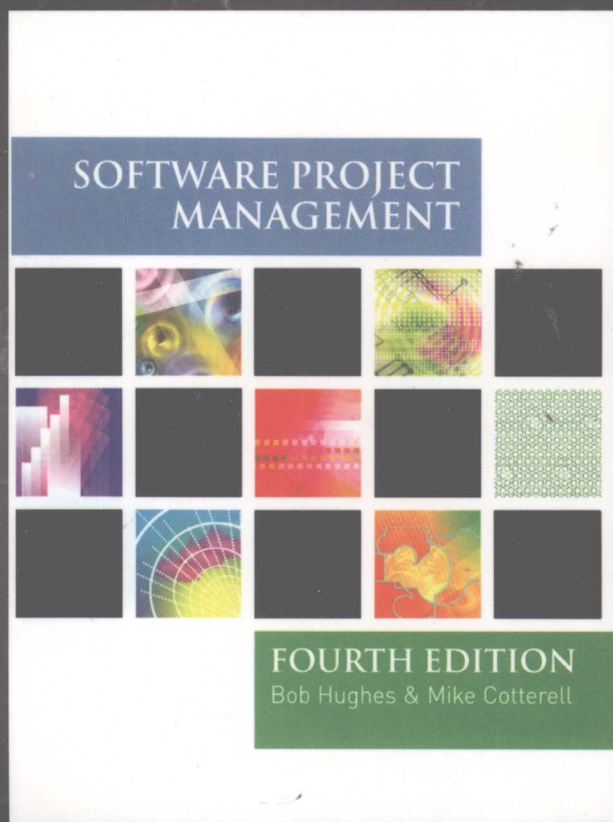
Mc
Graw
Hill

计 算 机 科 学 丛 书

原书第4版

软件项目管理

(英) Bob Hughes Mike Cotterell 著 廖彬山 王慧 译 周伯生 审校



Software Project Management Fourth Edition



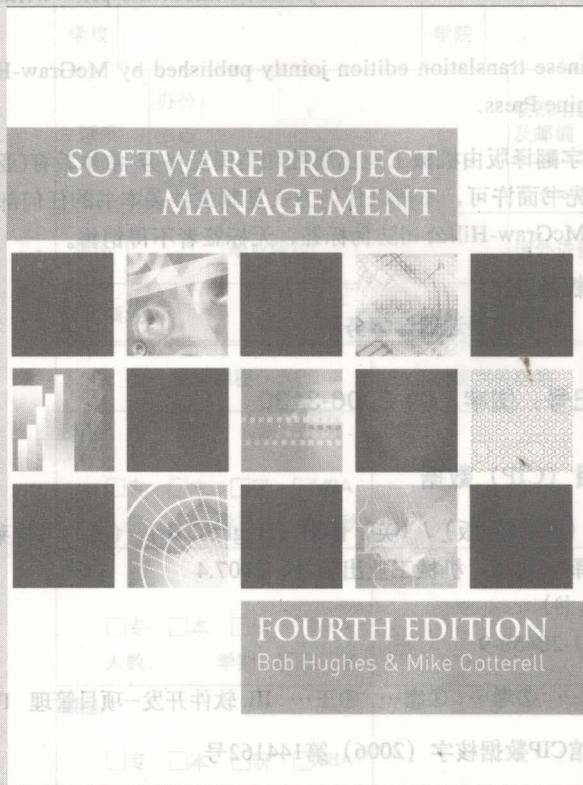
机械工业出版社
China Machine Press

TP311.52
X657.1

计算机科学丛

软件项目管理

(英) Bob Hughes Mike Cotterell 著 廖彬山 王慧 译 周伯生 审校



Software Project Management Fourth Edition

机械工业出版社
China Machine Press

本书是经典的项目管理课程教材。全书从项目的角度，采用步进式策划方法逐一分析了软件开发的各个环节，并附有具体实例。主要内容包括：项目评估、选择合适的项目方法、软件工作量估计、活动策划、风险管理、资源分配、监督与控制、管理合同、管理人员和组织群组、软件质量等。第4版新增了国际间项目的协作开发、软件外包、极限编程等新技术和新主题的介绍。书中提供了大量的辅助练习，并在附录中给出了练习答案。

本书不仅适合作为计算机及相关专业的本科生和研究生的教材，而且适合于软件项目管理人员和软件开发人员阅读，还特别适合作为BCS/ISEB专业考试的参考书。

Bob Hughes and Mike Cotterell: Software Project Management, Fourth Edition (ISBN 0-07-710989-9).

Copyright © 2006 by The McGraw-Hill Companies, Inc.

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签，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：01-2006-3130

图书在版编目(CIP)数据

软件项目管理(原书第4版)/(英)休斯(Hughes, B.), (英)考特莱尔(Cotterell, M.)
著; 廖彬山, 王慧译. —北京: 机械工业出版社, 2007.4

(计算机科学丛书)

ISBN 978-7-111-20458-9

I. 软… II. ①休… ②考… ③廖… ④王… III. 软件开发—项目管理 IV. TP311.52

中国版本图书馆CIP数据核字(2006)第144162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: 王玉

北京慧美印刷有限公司印刷 · 新华书店北京发行所发行

2007年10月第1版第2次印刷

184mm × 260mm · 21印张

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

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的

教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件: hzjsj@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范 明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅 宏	程 旭	程时端	谢希仁
裘宗燕	戴 葵			

致谢

自1992年出版以来，我们已得到了许多人的帮助。他们是：Ken Tanson, Chris Claire, David How, Martin Campbell, Kelly, Barbara Kitchenham (它们使我们使用第2章中给出的项目数措辞)，第六本Charmaster软件使用的Paul Rindorf, Robyn Lawrie, David Gammal, John David Herton(后四位为第10章提供了资料)，David Purves, David Wayne, Dick Searles, John Pyman, Jim Watson, Mary Shepherd, Susan Chalm, David Wilson, David Farthing, Charlie

第4版前言

迄今为止，ICT（信息和通信技术）项目开发失败的风险仍然像10年前本书第1版出版时那样受到关注。本书各版本关注的重点都在于为ICT实践人员提供最好的项目管理基础实践，作为他们专业基础教育的一部分。我们希望现在这个版本可以为此提供更多的帮助。本书的主要读者对象是计算机与信息系统专业的本科和研究生。

我们决定再版这本书的原因在于书中包含的软件项目开发技术在近些年又有了长足的发展。以下列出了这些新的技术发展：

- 电子通信的发展使得身处异地的人员可以方便地参与同一个项目的工作。
- 基于以上这一点，软件开发向第三世界尤其是印度的外包有所增加。本书也出现了印度和中国的译本。
- 很多与软件产品质量及其过程质量评估相关的、新的和修订的国际标准大量出现。这应该在“离岸”（offshoring）的前提之下来看待，说明了要交付的软件产品的质量正在受到关注。
- 人们对敏捷方法尤其是极限编程（XP）的兴趣正在逐步增强。同样，这也可以在“离岸”软件开发的前提之下看待。西方的软件开发人员需要找到可以与廉价的开发者进行竞争的方式，比如对用户需求的更加重视、开发速度的提高，像XP所能提供的那样。

除了外部的开发者，使用本书作为教材的许多讲师对本书的内容也进行了评审。他们的建议也在新版本中有所体现。例如，第7章中Goldratt的关键链管理技术，因为时间和版面的关系，有一些技术不能加入到书中，但是希望读者通过阅读相关资料了解这些方法。

以前在附录中的一些主题，如IOS/IEC 12207和项目群管理现在加入到了正文之中。各种不同的项目管理标准和“知识体系”（BOK）仍然安排在附录之中。

在准备本书的新版本期间，我们在第7章风险管理中使用了Abdullah Al Shehab和David I. Shepherd提供的材料。David还在挣值分析方面提出了建议。在这里要特别感谢他们，还有我所在的布赖顿大学的同事，尤其是Marian Eastwood和Joe Wilkinson的建议和支持。感谢本书的评审人员，他们付出了时间和精力，为本书提供了有价值的反馈。

本书以前的各版本为正在努力解决“现实世界”中的IT开发问题的学生们做出了贡献，我们期待着那样的贡献——现在应该有更大的贡献。

致谢

自1995年筹划撰写前三版以来，我们已得到了许多人的帮助。他们是：Ken I'Anson, Chris Claire, David Howe, Martin Campbell Kelly, Barbara Kitchenham（允许我们使用第5章中给出的项目数据集），墨尔本 Charismatek软件度量的Paul Radford和Robyn Lawrie, David Garmus和David Herron(后面四位为第10章提供了资料), David Purves, David Wynne, Dick Searles, John Pyman; Jim Watson, Mary Shepherd, Sunita Chulani, David Wilson, David Farthing, Charlie

Svahnberg, Henk Koppelaar和Ian McChesney。

我们还要感谢McGraw-Hill公司的团队。特别要感谢Karen Mosman，他是一个值得尊敬的人。作为我们的编辑，在这充满活力而又有点混乱（对于学术界来讲）的图书出版界，他是一个坚定的始终给予我们支持的联系人。然而，我们也不能忘记Dave Hatter，本书前一版的编辑，之前就职于Thomson 国际出版集团，现在就职于McGraw-Hill公司，并祝他在Essex的退休生活快乐。

我们还要感谢以下评审人员对本书的出版在不同阶段做出的贡献，他们是：

Dave Farthing, University of Glamorgan, UK

Peter Lee, University of Newcastle-on-Tyne, UK

Robert Macredie, Brunel University, UK

Owen Johnson, University of Leeds, UK

Rory V.O'Connor, Dublin City University, Ireland

Andrew Rae, University of Paisley, UK

Torbjørn Strøm, Buskerud University College, Norway

A.S.M. Sajeev, University of New England, Armidale, Australia

Phil Thompson, Coventry University, UK

目 录

出版者的话	
专家指导委员会	
第4版前言	
第1章 软件项目管理引言	1
1.1 引言	1
1.2 为什么软件项目管理很重要	1
1.3 什么是项目	2
1.4 软件项目和其他类型的项目	3
1.5 合同管理和技术项目管理	4
1.6 软件项目管理覆盖的活动	4
1.7 计划、方法和方法学	6
1.8 软件项目的分类方法	7
1.9 什么是管理	8
1.10 软件项目的问题	9
1.11 设定目的	10
1.12 项目相关人员	11
1.13 业务案例	12
1.14 需求规格说明	12
1.15 管理控制	13
1.16 小结	14
附件1 可行性研究内容列表	14
附件2 项目计划内容列表	15
1.17 进一步的练习	15
第2章 步进式方法：项目策划概述	17
2.1 步进式项目策划概述	17
2.2 第0步：选择项目	20
2.3 第1步：标识项目的范围和目的	20
2.4 第2步：标识项目的基础设施	21
2.5 第3步：分析项目的特征	23
2.6 第4步：标识项目的产品和活动	25
2.7 第5步：估计每个活动的工作量	28
2.8 第6步：标识活动的风险	30
2.9 第7步：分配资源	30
2.10 第8步：评审/发布计划	32
2.11 第9步和第10步：执行计划并进行 较低层次的策划	32
2.12 小结	33
2.13 进一步的练习	33
第3章 项目群管理和项目评价	35
3.1 引言	35
3.2 项目群管理	35
3.3 管理项目群内的资源分配	38
3.4 项目群管理策略	39
3.5 创建项目群	39
3.6 辅助项目群管理	42
3.7 收益管理	43
3.8 单个项目评估	45
3.9 技术评估	45
3.10 成本效益分析	45
3.11 现金流预测	46
3.12 成本效益评价技术	47
3.13 风险评价	52
3.14 小结	57
3.15 进一步的练习	57
第4章 选择合适的项目方法	59
4.1 引言	59
4.2 选择技术	59
4.3 技术计划内容清单	63
4.4 过程模型的选择	64
4.5 结构与交付速度	65
4.6 瀑布模型	65
4.7 V过程模型	66
4.8 螺旋模型	67
4.9 软件原型开发	67
4.10 分类原型的其他方法	69
4.11 增量式交付	71

4.12 动态系统开发方法	74	6.18 进一步的练习	127
4.13 极限编程	75	第7章 风险管理	129
4.14 管理迭代过程	77	7.1 引言	129
4.15 选择最合适的过程模型	78	7.2 风险	129
4.16 小结	78	7.3 风险分类	130
4.17 进一步的练习	78	7.4 处理风险的框架	133
第5章 软件工作量估计	81	7.5 风险识别	133
5.1 引言	81	7.6 风险评估	136
5.2 在何处进行估计	82	7.7 风险策划	139
5.3 估计过高和估计过低的问题	83	7.8 风险管理	140
5.4 软件估计基础	85	7.9 评价进度风险	142
5.5 软件工作量估计技术	86	7.10 应用PERT技术	143
5.6 专家判断	88	7.11 蒙特卡罗仿真	147
5.7 类比估计	88	7.12 关键链概念	147
5.8 Albrecht功能点分析	88	7.13 小结	150
5.9 Mark II功能点	91	7.14 进一步的练习	150
5.10 COSMIC全功能点	93	第8章 资源分配	153
5.11 面向过程代码的方法	94	8.1 引言	153
5.12 COCOMO: 一个参数模型	95	8.2 资源的性质	153
5.13 小结	100	8.3 标识资源需求	155
5.14 进一步的练习	100	8.4 资源调度	157
第6章 活动策划	103	8.5 创建关键路径	160
6.1 引言	103	8.6 计算成本	160
6.2 活动策划的目的	103	8.7 特定的细节	161
6.3 何时计划	104	8.8 发布资源进度表	162
6.4 项目进度表	104	8.9 成本进度	164
6.5 项目和活动	106	8.10 调度顺序	165
6.6 确定活动的次序和进度	109	8.11 小结	166
6.7 网络策划模型	111	8.12 进一步的练习	166
6.8 网络模型的公式化表示	112	第9章 监督与控制	167
6.9 增加时间维	115	9.1 引言	167
6.10 正向遍历	115	9.2 创建框架	167
6.11 反向遍历	116	9.3 收集数据	170
6.12 标识关键路径	118	9.4 进展可视化	172
6.13 活动缓冲期	119	9.5 成本监督	175
6.14 缩短项目周期	119	9.6 挣值分析	176
6.15 标识关键活动	120	9.7 优先级控制	180
6.16 活动-箭头网络	120	9.8 使项目返回目标	181
6.17 小结	126	9.9 变更控制	182

9.10 小结183

9.11 进一步的练习184

第10章 管理合同185

10.1 引言185

10.2 获取和供应软件产品的ISO 12207
方法186

10.3 供应过程188

10.4 合同的种类189

10.5 合同部署的阶段193

10.6 典型的合同条款197

10.7 合同的管理200

10.8 验收201

10.9 小结201

10.10 进一步的练习201

第11章 管理人员和组织群组203

11.1 引言203

11.2 理解行为203

11.3 组织行为：背景205

11.4 为工作选择合适人选206

11.5 用最好的方法进行教育207

11.6 激励工作热情208

11.7 Oldham-Hackman工作特征模型210

11.8 在组内工作211

11.9 成为群组211

11.10 决策制定213

11.11 领导能力214

11.12 组织结构216

11.13 分散的或虚拟团队218

11.14 文化的影响221

11.15 压力223

11.16 健康和安 全224

11.17 小结225

11.18 进一步的练习225

第12章 软件质量227

12.1 引言227

12.2 软件质量在项目策划中的位置227

12.3 软件质量的重要性228

12.4 定义软件质量229

12.5 ISO 9126230

12.6 实用的软件质量度量235

12.7 产品与过程质量管理236

12.8 外部标准237

12.9 有助于提高软件质量的技术241

12.10 质量计划245

12.11 小结246

12.12 进一步的练习246

第13章 小型项目249

13.1 引言249

13.2 学生项目的一些问题249

13.3 项目计划的内容251

13.4 小结253

附录A PRINCE2 概述255

附录B BS 6079: 1996概述265

附录C 练习答案271

进一步阅读材料303

索引307

目的

学完本章后，你将能够：

- 定义软件项目管理的范围。
- 了解软件项目经理的问题和担忧。
- 定义软件项目的各个阶段。
- 解释管理作用的要素。
- 意识到需要精心策划、监督和控制。
- 识别项目相关人员以及他们的目的。
- 定义项目的成功标准。

1.1 引言

本书是关于“软件项目管理”的。在第1章中，我们要解决的重点问题是软件项目管理是否真的区别于其他类型的项目管理。为了回答这个问题，我们需要了解软件项目的策划、监督和控制的关键思想。我们将会看到，项目主要是为了满足目的。像其他项目一样，生产软件的项目只有在满足实际需求的时候才有价值。因此，需要检查怎样才能识别项目相关人员以及他们的目的。在认识了这些目的以后，保证满足这些目的是成功的项目管理的基础。然而，除非已经有了准确的信息，否则这些工作都无法进行，因而要研究如何获得准确的信息。

1.2 为什么软件项目管理很重要

本书的读者包括软件工程和计算机科学方向的学生以及从事业务信息系统学习的学生。从事技术工作的学生在学习那些与编程无关的课程时，往往会没有耐心。那么熟悉项目管理的重要性又是什么呢？

首先是钱的问题。很多的经费都用来作为ICT项目的赌注。2002年至2003年，英国政府花费在ICT项目上的经费高于公路系统上的经费（大约是23亿英镑比14亿英镑）。其中花费最多的部门是就业和退休保障部，他们在ICT项目上的花费为8亿英镑。如果不能很好地管理ICT项目，那么能用在其他方面（例如养老保险）的资金就会降低。

非常不幸的是，并不是每个项目都很成功。在2003年的一份分析报告中，美国Standish Group通过分析13 522个项目得出结论：只有1/3的项目是成功的，82%的项目延期，43%的项目超出预算。

通常可以找到的导致项目缺陷的原因都与项目管理有关。例如，英国的

本段信息摘自
2004年11月的英国
国家审计官方
报告：*Improving
IT Procurement.*

国家审计署指出，在众多导致项目失败的原因中包括：缺乏项目管理和风险管理的能力和有效的方法。

1.3 什么是项目

根据Longman Concise English Dictionary(1982),“项目”的字典定义包括：“一个特定的计划或设计”；“一个已计划的任务”；“一项大型任务，如公共事业方案”。

字典中在定义项目时清楚地强调项目是已计划的活动。

把项目定义为已计划的活动时假定：在很大程度上，可以在项目开始之前决定怎样执行任务。但可能有一些相当困难的试探性项目。策划在本质上就是在做某些事之前的周密考虑，甚至有一些不太确定的、但值得做的项目，只要接受了，就要周密考虑，但其最终计划有临时性和冒险性的因素。其他如与日常维护相关的活动，可能会重复很多次，以至于每一个有关人员都准确地知道需要做些什么。在这种情况下，尽管仍然需要编写文档化的规程，以保证工作的一致性和帮助新员工熟悉其工作，但策划似乎显得不是很必要的了。

从传统的项目管理中受益最多的活动类型可能是两种极端情况之间的活动类型（参见图1-1）。



图1-1 很可能从项目管理中受益的活动

非常规项目和常规项目之间的界限并不明确。第一次做一项常规工作就像在做个项目。而开发一个与你以前开发的项目类似的系统，则这个项目会有大量的常规成分。

区分项目性质的特征有以下几点：

- 涉及非常规任务。
- 需要策划。
- 要满足特定的目标或要创造特定的产品。
- 项目有预先确定的时间跨度。
- 为别人工作而不是为自己工作。
- 工作涉及到几个学科。
- 工作分几个阶段进行。
- 可用于项目的资源是受到限制的。
- 项目庞大或者复杂。

以上因素在任务中越多，该任务就越困难。项目规模是一个特别重要的因素，管理一个拥有20个员工的项目要比管理只有10个人的项目棘手得多。

2

项目群管理通常用来协调当前工作的活动。

大项目的困难程度不成比例地增加，额外的沟通导致其比小项目在管理上需要增加不止两倍的工作量。本书的示例和练习通常涉及较小项目，从学习的角度考虑这样做便于管理，而其中所讨论的技术和问题与大型项目是类似的。

练习1.1 考虑下列问题：

- 编辑一版报纸。
- 在火星上投放机器人设备搜索生命迹象。
- 准备结婚。
- 改进财务计算机系统以处理通用的欧元问题。
- 开发优秀人机界面的研究项目。
- 对用户使用计算机系统时出现的问题的原因进行调查。
- 为计算领域学生分配第二年的编程任务。
- 为一台新计算机编制操作系统。
- 在组织中安装新版本的字处理软件包。

其中有些用“项目”描述可能比其他更合适。把它们按照最接近你的构成项目的概念的顺序排列起来，越在上面，越符合“项目”的概念。对其中的每一条，描述与其上一条在作为项目上的不同点。

这个练习没有惟一正确的答案，但是在本书的末尾，可以找到这个练习以及其他练习的可能解答。

1.4 软件项目和其他类型的项目

很多通用的项目管理技术可以用于软件项目管理，但Fred Brooks指出，与其他项目的产品相比，软件产品的产品具有某些不同的特征。

不可见性 一个像桥或路这样的有形制品的构造过程的进展是可以实际看到的。而对于软件，其进展是不能立即看到的。理解软件项目管理的一种方法是使原来不可见的过程可视化。

复杂性 对于每一美元、每一英镑或每一欧元在软件产品上是如何花费的，要比其他工程制品更加复杂。

一致性 “传统的”工程师通常使用物理系统以及像水泥和钢铁这样的物理材料来工作。这些物理系统可能有一定的复杂性，但都服从某些一致的物理定律。软件开发者必须与客户需求保持一致。这不仅是因为从事该工作的人员可能不是同一个人，而且对于一个组织来说，还由于集体记忆会有差错、内部交流不够畅通、决策也会失误，这些都要开发者来协调解决。

灵活性 软件可以被方便地改变，通常认为这是软件的长处之一。然而，这意味着软件系统一旦与一个物理的或组织的系统相连接，在有必要时，就期望改变软件来适应其他组件，而不是改变其他组件来适应软件。因而相对于其他组件来说，软件系统很可能要经常变更。

Brooks, F. P. (1987), “没有银弹：软件工程的本质和事故”。这篇文章收录在《人月神话》周年纪念版上，Addison Wesley, 1995。

1.5 合同管理和技术项目管理

许多组织都将ICT产品的开发外包给专业开发人员。在这种情况下，客户方往往会派一名“项目经理”来指导合同的执行。项目经理会授权承包商作出许多面向技术的决策。例如，只要整个项目能在预算内按时完成，项目经理就不会关心编写单独的软件组件的工作量估计问题。从供应商的角度考虑，需要有项目经理来关注更多的技术管理问题。本书着重讨论这些“技术型”项目经理关注的问题。

1.6 软件项目管理覆盖的活动

软件项目不仅仅关心软件的实际编写问题。事实上，当购买一个“商用产品”作为软件应用时，可能就没有这类编写软件的工作。但这仍然是一个基本的软件项目，因为还有众多的其他因素与这类项目相关联。

产生一个新系统通常需要三个连续的步骤（参见图1-2）。

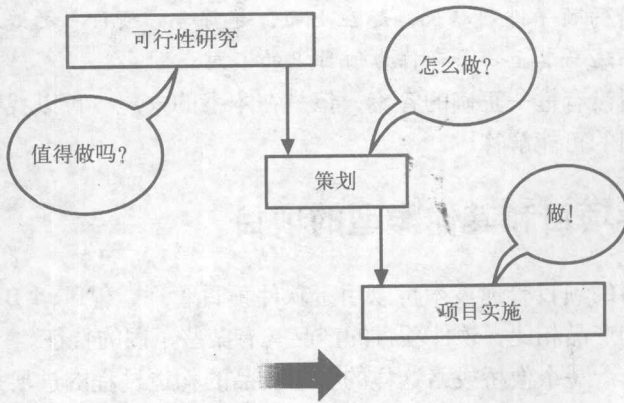


图1-2 可行性研究、策划和项目实施

1) **可行性研究** 这是一项调查研究，旨在判断一个预期的项目是否值得开始——即存在有效的业务案例。在研究期间，要收集有关所建议的应用需求方面的信息。至少在最开始，需求的引出是复杂且困难的。客户和其他相关人员可能对他们期待解决的问题以及要达到的目标有很多想法，但是却不了解解决这些问题和达到这些目标的方法。新系统可能的开发和运行成本及其效益都需要进行估计。对一个大型系统而言，可行性研究本身可以作为一个项目来处理，并且有它自己的策划子阶段。可行性研究是战略策划的一部分，用于考察所有潜在的软件开发任务并对其进行排序。有时当要把一组项目作为项目群（programme）开发时，组织需要制定一个方针。

2) **策划** 如果可行性研究的结果指出预期的项目可行，就可进入项目的策划阶段。然而，对于大型项目而言，并不是在刚开始时就能正确地制定出全部的详细计划。而是可以为整个项目制定一个概要计划，并为第一阶段制定一个详细计划。后面各阶段的详细计划要在进入该阶段时再做。因为只有

在关于项目分析和
技术策划的第4章中，
查阅某些可以代替的
生命周期。

4

本章的附件1
给出了可行性
研究的内容提纲。

第3章进一步
讨论了项目群
管理的各个方面。

本书附录A描
述的PRINCE2方
法用阶段化方法
进行策划。

临近该后续阶段，才有更详细的可用来制定计划的准确信息。

3) 项目实施 现在可以实施项目了。项目实施往往包括设计和实现两个子阶段。初做项目策划的学生经常发现很难把策划和设计分开，并且这两者之间的界限也往往比较模糊。本质上，设计是对项目要生产的产品的准确形式的思考和制定决策。对软件来说，设计与软件外观（即用户界面）或者内部结构有关。策划则规定了为生产产品所必须执行的活动。由于在大多数细节的层次上，策划决策要受到设计决策的影响，因此策划和设计可能会被混淆。例如，假如一个软件产品有五个主要组件，那么很可能要由五组活动来制造它们。

图1-3表示了国际标准ISO 12207中定义的软件开发活动的典型顺序。读者可能会发现其中一些是关于系统（system）的，而另一些是关于软件（software）的。通常，软件开发仅是整个项目中的一部分。例如，在一个项目中需要开发软件，同时也需要建立ICT底层、设计用户的工作以及针对新的软件对客户进行培训。这些加起来是一个系统，而软件是其中的一部分。

本章附件2有
策划内容的大纲。

图1-3建议
这些阶段必须严格
按顺序做。我们
在第4章将看到，
可以采用其他的
迭代方法。不过，
这儿列出的实际
活动仍然要做。

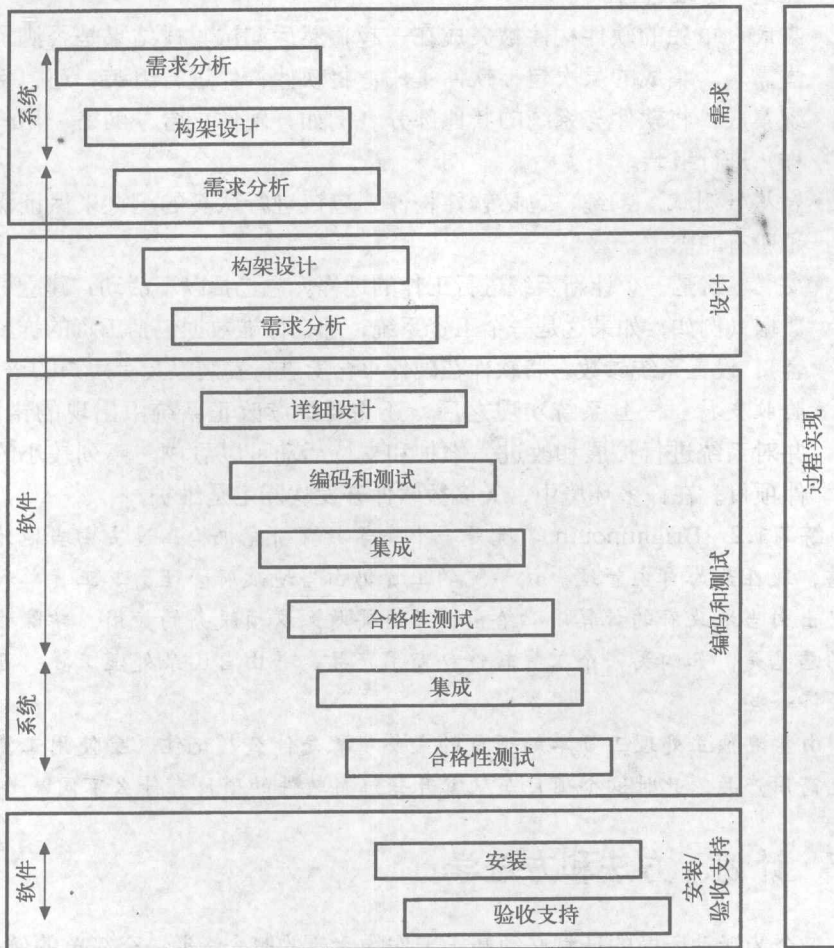


图1-3 ISO 12207软件开发生命周期

6

- **需求分析** 以需求引导作为其开始，需求引导的目的是发掘潜在客户、他们的经理以及雇员对新系统在功能和性能上的需求。这些将作为一个整体与系统建立关联。例如，质量需求可能包括：用户可以在一定的时间内完成一个事务。在这个例子中，事务完成时间会受到人员操作速度、软件和硬件性能等因素的影响。这些“面向客户”的需求需要被转换成新系统开发者可以使用的技术需求。
- **构架设计** 将需求映射到系统构件。在系统层需要做出决策：新系统中哪一些过程由用户提供，哪一些由电脑处理。这些关于系统构架（system architecture）的设计是软件需求开发的输入。然后是构架设计的第二步：将软件需求映射到软件构件。
- **详细设计** 将每个软件构件分解成可以单独进行编码和测试的软件单元。这些单元的详细设计将分别进行。
- **编码和测试** 这可能涉及用过程语言（如C# 或 Java）来编写代码，或者可能涉及使用像Microsoft Access这样的应用建造工具。单元测试也在这一阶段进行。
- **集成** 单独的软件构件被集成在一起，然后测试集成结果是否满足整体需求。集成的层次包括软件集成：将软件构件进行组装，还包括系统集成：将软件与系统的其他部分（例如，硬件平台、网络、用户过程）进行组装。
- **合格性测试** 系统，包括软件构件，应该进行认真的测试来保证了所有的需求。
- **安装** 这是一个让新系统进行工作的过程。它包括以下活动：建立标准数据（例如，如果这是一个工资系统，那么标准数据包括工资的详细内容）、设置系统参数、将软件在硬件平台上进行安装以及对用户的培训。
- **验收支持** 一旦系统实现之后，还需要继续改正系统中出现的错误，并对系统进行扩展和改进。维护和支持活动可以看成一系列较小的软件项目。在许多环境中，大多数软件开发实际上是维护。

练习1.2 Brightmouth 学院是一个高等教育研究机构，过去由当地政府管辖，现在改为自主管理。该学院的工资仍由当地政府管理，其工资单和其他支出由当地政府的计算中心给出，政府要收取这项服务的费用。学院管理方的意见是，如购买一个工资软件包商用产品，并由自己来处理工资，可能要经济一些。

由学院独立处理工资单的项目的主要步骤是什么？记住，要使用工资软件包商用产品，此时这个项目与从零开始编写软件的项目有什么不同？

7

1.7 计划、方法和方法学

一个为活动所做的计划必须基于工作的方法的概念。举一个简单的例子，如果让你去测试某个软件，即使你对要测试的软件一无所知，也可以设想将