

软件项目管理

(原书第3版)

Software Project Management

(Third Edition)

(英) Bob Hughes 著
Mike Cotterell

周伯生 廖彬山 任爱华 等译



机械工业出版社
China Machine Press

软件项目管理

(原书第3版)

Software Project Management

(Third Edition)

(英) **Bob Hughes** 著
Mike Cotterell

周伯生 廖彬山 任爱华 等译



机械工业出版社
China Machine Press

本书从项目的角度,采用步进式策划方法对软件开发的各个环节逐一进行分析,并附有具体实例。主要内容包括:项目评价、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。另外,还提供了大量的辅助练习,并在附录 F 中给出了练习答案。

本书是项目管理课程的入门教材,不仅适用于计算机及相关专业的大学本科生和研究生,而且适于软件项目管理人员、软件开发人员阅读,同时还特别适合作为 BCS/ISEB 专业考试的参考书。

Bob Hughes and Mike Cotterell: Software Project Management, Third Edition (ISBN 0-07-709834-X).
Copyright © 2002 by McGraw-Hill International (UK) Limited.

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 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。

本书版权登记号:图字:01-2002-4185

图书在版编目(CIP)数据

软件项目管理(原书第3版)/(英)休斯(Hughes, B.)等著;周伯生等译. - 北京:机械工业出版社, 2004.7

(软件工程技术丛书 项目管理系列)

书名原文: Software Project Management, Third Edition

ISBN 7-111-14148-2

I. 软… II. ①休…②周… III. 软件开发-项目管理 IV. TP311.52

中国版本图书馆 CIP 数据核字(2004)第 021112 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:迟振春

山东高唐印刷有限责任公司印刷·新华书店北京发行所发行

2004 年 7 月第 1 版第 1 次印刷

787mm×1092mm 1/16·22 印张

印数:0 001-4 000 册

定价:36.00 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

前 言

对第3版的编作者起初是犹豫的。特别是，如果新版本意味着教师必须对引用本书特定章节的材料进行修改，那么以本书为教材的教师肯定不愿采纳新版本。不过，软件项目管理学科的确有了许多新进展，我们必须重视，特别是：

- 软件项目的更“敏捷”方法，如动态系统开发方法（DSDM）和极限编程（XP）。
- 2000年提出的新版的BS EN ISO 9000系列标准和指南。
- 在项目群管理语境中，需要增强项目管理的意识。
- 美国项目管理协会（PMI）和英国项目管理协会（APM）制定的新版的项目管理知识体系。
- 在项目环境内日益需要考虑健康和安全问题，特别要考虑压力问题。

对本书的其它修改源自于在教学中摸索的结果。特别在前几章中，对经常引起误解的内容做了修订并增加了一些内容。例如，增加了项目计划内容列表作为第1章的附件，而在第2章增加了甘特图的例子。这是因为我们发现让学生对最终产品是什么样子有个概念是有帮助的，即使他们还了解创建产品的步骤。

另外，还根据需要删掉了一些材料。例如，由于增加了关于项目群管理的一个附录，考虑到内容的冗余，删掉了第2版中关于Euromethod的内容。自前一版出版以来，Bob Hughes的*Practical Software Measurement*（实用软件度量）已经出版，因此本书关于目标/问题/度量元（GQM）范式的章节已经删掉。对GQM感兴趣的读者最好从专门介绍度量的书中查阅更完整的内容。

自1995年开始在编写本书各个版本的过程中，我们得到了许多人的帮助。这些人包括：Ken I'Anson、Chris Claire、David Howe、Martin Campbell Kelly、Barbara Kitchenham（允许使用第5章中给出的项目数据集）、Charismatek软件度量元的Paul Radford和Robyn Lawrie、Melbourne、David Garmus和David Herron（最后4章，第10章的所有材料），David Purves、David Wynne、Dick Searles、John Pyman、Jim Watson、Mary Shepherd、Sunita Chulani、David Wilson、David Farthing、Charlie Svahnberg、Henk Koppelaar、Ian McChesney。

我们还要感谢McGraw-Hill的工作人员。特别要感谢汤姆森国际出版社（后来在McGraw-Hill）的编辑Dave Hatter，他激励我们编写本书第3版。

第1版是献给那些现在奋斗在IT开发领域的我们以前的学生的，这一版仍然献给他们。

目 录

前言	
第1章 软件项目管理引言	1
1.1 引言	1
1.2 什么是项目	1
1.3 软件项目和其它类型的项目	3
1.4 合同管理和技术项目管理	3
1.5 软件项目管理覆盖的活动	3
1.6 计划、方法和方法学	5
1.7 软件项目的分类方法	6
1.7.1 信息系统与嵌入式系统	6
1.7.2 目的与产品	6
1.8 什么是管理	7
1.9 软件项目的问题	7
1.10 设定目的	9
1.10.1 子目的和目标	9
1.10.2 有效性度量	10
1.11 项目相关人员	10
1.12 业务案例	11
1.13 需求规格说明	11
1.14 管理控制	12
1.15 小结	13
附件1 可行性研究内容列表	13
附件2 项目计划内容列表	14
1.16 进一步的练习	14
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	22
2.5 第3步：分析项目的特征	24
2.6 第4步：标识项目的产品和活动	26
2.7 第5步：估计每个活动的工作量	30
2.8 第6步：标识活动的风险	31
2.9 第7步：分配资源	32
2.10 第8步：评审/发布计划	33
2.11 第9步和第10步：执行计划 并进行较低层次的策划	34
2.12 小结	34
2.13 进一步的练习	34
第3章 项目评价	37
3.1 引言	37
3.2 战略评估	37
3.2.1 项目群管理	37
3.2.2 业务量管理	39
3.3 技术评估	39
3.4 成本效益分析	40
3.5 现金流预测	41
3.6 成本效益评价技术	42
3.6.1 净利润	42
3.6.2 回收期	43
3.6.3 投资回报率	43
3.6.4 净现值	43
3.6.5 内部回报率	46
3.7 风险评价	48
3.7.1 风险标识和排等级	48
3.7.2 风险和净现值	48
3.7.3 成本效益分析	49
3.7.4 风险剖面分析	50
3.7.5 使用决策树	50
3.8 小结	52
3.9 进一步的练习	53
第4章 选择合适的项目方法	55
4.1 引言	55
4.2 选择技术	55
4.2.1 将项目标识为目的驱动 或产品驱动	57

4.2.2 分析其它项目特征	57	5.5 软件工作量估计技术	82
4.2.3 标识高级别项目风险	58	5.5.1 由底向上估计	83
4.2.4 考虑与实现有关的用户需求	59	5.5.2 自顶向下法和参数模型	83
4.2.5 选择通用的生命周期方法	59	5.6 专家判断	84
4.3 技术计划内容清单	60	5.7 类比估计	84
4.4 过程模型的选择	61	5.8 Albrecht 功能点分析	85
4.5 结构与交付速度	61	5.9 Mark II 功能点	88
4.6 瀑布模型	62	5.10 对象点	90
4.7 V 过程模型	63	5.11 面向过程代码的方法	91
4.8 螺旋模型	64	5.12 COCOMO: 一个参数模型	92
4.9 软件原型开发	65	5.13 小结	97
4.10 分类原型的其它方法	66	5.14 进一步的练习	97
4.10.1 要从原型中学到什么	66	第 6 章 活动策划	99
4.10.2 原型要做到什么程度	66	6.1 引言	99
4.10.3 哪些要进行原型化	67	6.2 活动策划的目的	99
4.11 在原型开发期间控制变更	67	6.3 何时计划	100
4.12 增量式交付	68	6.4 项目进度表	101
4.12.1 优点	68	6.5 项目和活动	101
4.12.2 缺点	69	6.5.1 定义活动	101
4.12.3 增量式交付计划	69	6.5.2 标识活动	103
4.12.4 标识系统目的	69	6.6 确定活动的次序和进度	107
4.12.5 创建开放的技术计划	70	6.7 网络策划模型	108
4.12.6 计划增量	70	6.8 网络模型的公式化表示	108
4.12.7 增量示例	71	6.8.1 构造优先网络	108
4.13 动态系统开发方法	71	6.8.2 描绘滞后活动	111
4.14 极限编程	73	6.8.3 吊床活动	111
4.15 管理迭代过程	73	6.8.4 标注约定	112
4.16 选择最合适的过程模型	75	6.9 增加时间维	112
4.17 小结	75	6.10 正向遍历	113
4.18 进一步的练习	75	6.11 反向遍历	114
第 5 章 软件工作量估计	77	6.12 标识关键路径	115
5.1 引言	77	6.13 活动缓冲期	117
5.2 在何处进行估计	79	6.14 缩短项目周期	117
5.3 估计过高和估计过低的问题	80	6.15 标识关键活动	118
5.4 软件估计基础	81	6.16 活动 - 箭头网络	118
5.4.1 需要历史数据	81	6.16.1 活动 - 箭头网络的规则和约定	118
5.4.2 工作的度量	81	6.16.2 使用虚活动	120
5.4.3 复杂性	82	6.16.3 描绘滞后的活动	120

6.16.4 给活动加标签	122	8.8 发布资源进度表	155
6.16.5 网络分析	123	8.9 成本进度	155
6.17 小结	125	8.10 调度顺序	158
6.18 进一步的练习	126	8.11 小结	159
第 7 章 风险管理	127	8.12 进一步的练习	160
7.1 引言	127	第 9 章 监督与控制	161
7.2 风险的本质	127	9.1 引言	161
7.3 风险的类型	129	9.2 创建框架	161
7.3.1 估计错误	130	9.2.1 责任	162
7.3.2 策划假设	130	9.2.2 进展评估	163
7.3.3 不可预计的事	130	9.2.3 设置检查点	163
7.4 管理风险	130	9.2.4 取快照	164
7.5 危险标识	132	9.3 收集数据	164
7.6 危险分析	134	9.3.1 局部完成报告	164
7.6.1 风险危害度	135	9.3.2 风险报告	165
7.6.2 设置风险优先级	136	9.4 进展可视化	167
7.6.3 风险缓解效果	137	9.4.1 甘特图	167
7.7 风险策划和控制	137	9.4.2 延迟图	167
7.8 评价进度风险	139	9.4.3 球形图	168
7.8.1 使用 PERT 评价不确定性的影响	139	9.4.4 时间线	169
7.8.2 使用期望周期	140	9.5 成本监督	170
7.8.3 活动标准偏差	140	9.6 挣值	171
7.8.4 满足目标的可能性	141	9.6.1 基线预算	172
7.8.5 计算每个项目事件的标准偏差	142	9.6.2 监督挣值	174
7.8.6 计算 z 值	142	9.7 优先级控制	175
7.8.7 转换 z 值为概率	142	9.8 使项目返回目标	177
7.8.8 PERT 的优点	143	9.8.1 缩短关键路径	177
7.8.9 蒙特卡洛仿真	143	9.8.2 重新考虑优先需求	177
7.9 小结	144	9.9 变更控制	178
7.10 进一步的练习	144	9.9.1 配置库管理员角色	178
第 8 章 资源分配	145	9.9.2 变更控制规程	179
8.1 引言	145	9.9.3 系统范围的变更	179
8.2 资源的性质	145	9.10 小结	179
8.3 标识资源需求	147	9.11 进一步的练习	180
8.4 资源调度	149	第 10 章 管理合同	181
8.5 创建关键路径	152	10.1 引言	181
8.6 计算成本	153	10.2 合同的种类	182
8.7 特定的细节	154	10.2.1 固定价格合同	183

10.2.2 时间和材料合同	183	11.7 Oldham-Hackman 工作特征模型	204
10.2.3 每单位固定价格合同	184	11.8 在组内工作	205
10.2.4 公开的投标过程	186	11.9 成为群组	205
10.2.5 受限制的投标过程	186	11.10 决策制定	207
10.2.6 已商定的规程	186	11.10.1 制定正确决策的心理障碍	208
10.3 合同部署的阶段	187	11.10.2 小组决策的制定	208
10.3.1 需求分析	187	11.10.3 正确的小组决策制定的 障碍	208
10.3.2 评估计划	188	11.10.4 减少小组决策制定的 缺点的措施	209
10.3.3 邀请投标	189	11.11 领导能力	209
10.3.4 评估提议	189	11.12 组织结构	211
10.4 典型的合同条款	191	11.12.1 正式的与非正式的结构	211
10.4.1 定义	191	11.12.2 层次化的方法	211
10.4.2 协议的形式	191	11.12.3 员工与生产线	212
10.4.3 供应的商品和服务	191	11.12.4 部门化	212
10.4.4 软件的所有权	191	11.12.5 非自我编程	212
10.4.5 环境	192	11.12.6 主程序员组	213
10.4.6 客户承诺	192	11.13 压力	213
10.4.7 验收规程	192	11.14 健康和安全	214
10.4.8 标准	192	11.15 小结	215
10.4.9 项目和质量管理	192	11.16 进一步的练习	215
10.4.10 时间表	193	第 12 章 软件质量	217
10.4.11 价格和付款方式	193	12.1 引言	217
10.4.12 其它法律上的需求	193	12.2 软件质量在项目策划中的位置	217
10.5 合同的管理	193	12.3 软件质量的重要性	219
10.6 提交和验收	194	12.4 定义软件质量	219
10.7 小结	195	12.4.1 产品操作质量因素	220
10.8 进一步的练习	195	12.4.2 产品修订质量因素	220
第 11 章 管理人员和组织群组	197	12.4.3 产品转变质量因素	220
11.1 引言	197	12.5 ISO 9126	221
11.2 理解行为	198	12.6 实用的软件质量度量方法	225
11.3 组织行为: 背景	199	12.6.1 可靠性	225
11.4 为该工作选择合适人选	200	12.6.2 可维护性	225
11.5 用最好的方法进行教育	202	12.6.3 可扩展性	225
11.6 动机	202	12.7 产品与过程质量管理	226
11.6.1 Taylorist 模型	202	12.8 外部标准	227
11.6.2 Maslow 的需要层次	203	12.8.1 BS EN ISO 9001:2000	227
11.6.3 Herzberg 的双因素理论	203		
11.6.4 动机的期待理论	204		

12.8.2	BS EN ISO 9001:2000	
	QMS 需求概述	228
12.8.3	能力过程模型	229
12.8.4	评估软件产品	230
12.9	有助于提高软件质量的技术	230
12.9.1	审查	231
12.9.2	Fagan 方法的普遍原理	232
12.9.3	结构化编程和净室软件开发	232
12.9.4	形式化方法	233
12.9.5	软件质量循环	233
12.9.6	质量循环解决的问题	233
12.9.7	质量循环的效率	234
12.10	质量计划	235
12.11	小结	235
12.12	进一步的练习	236
第 13 章	小型项目	237
13.1	引言	237
13.2	学生项目的一些问题	237
13.2.1	使用不熟悉的工具	237
13.2.2	不确定的设计需求	238
13.2.3	不完整的系统	238
13.2.4	缺少客户对项目的承诺	239
13.3	项目计划的内容	239
13.4	小结	242
附录 A	PRINCE 2 概述	243
附录 B	BS 6079:1996 概述	253
附录 C	项目群管理	259
附录 D	ISO 12207 概述	271
附录 E	项目管理知识体系	281
附录 F	练习答案	289
	进一步阅读材料	321
	索引	327

第 1 章

软件项目管理引言

目的

学完本章后，你将能够：

- 定义软件项目管理的范围。
- 区分软件开发项目和其它类型的开发项目。
- 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。
- 认识项目相关人员、他们的目的以及定义成功达到这些目的的方法。

1.1 引言

什么是软件项目管理？它与普通意义上的管理有什么不同？软件项目管理与其它类型的项目管理有什么不同？要回答这些问题，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。生产软件的项目只有在满足实际需要时才有价值。因此，需要检查怎样才能认识项目相关人员以及他们的目的。在认识了这些目的之后，保证满足这些目的是一个成功项目的基础。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

1.2 什么是项目

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。其它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必要的了。

根据 *Longman Dictionary of the English Language* (1991)， “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”。

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）。

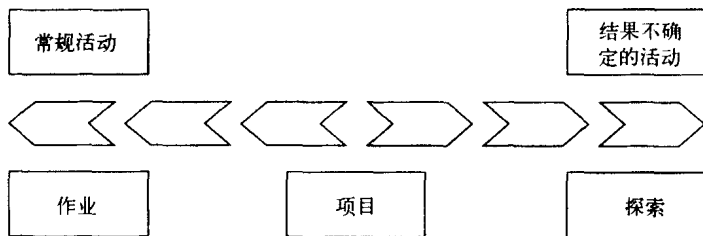


图 1-1 很可能从项目管理中受益的活动

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一个项目。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 200 个员工的项目要比管理只有两个人的项目棘手得多。本书的示例和练习通常涉及较小型的项目，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。
- 建立信道通道。
- 准备结婚。
- 改进财务计算机系统以处理通用的欧币问题。
- 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。

项目群管理通常用来协调几个并发作业的活动（参见附录 C）。

其中有些用“项目”描述可能比其它更合适。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它练习的可能解答。

1.3 软件项目和其它类型的项目

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，与其它项目的产品相比，软件项目的产品具有某些不同的特征。

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。

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。

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，要比其它工程制品更加复杂。

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它组件，而不是改变其它组件来适应软件。因而相对于其它组件来说，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。

3

1.4 合同管理和技术项目管理

许多组织都将 IT 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。

1.5 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

在关于项目分析和
技术策划的第4章中，
查阅某些可以替代的
生命周期。

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，因为还有众多的其它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）。

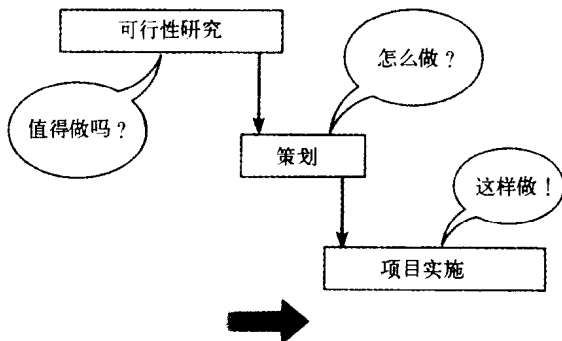


图 1-2 可行性研究、策划和项目实施

4

本章附件 1 给出了可行性研究的内容的提纲。

本书附录 C 有关于项目群管理的简短描述。

本书附录 A 描述的 PRINCE 2 方法用阶段化方法进行策划。

本章附件 2 有策划内容的大纲。

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。在研究期间，要收集有关所建议的应用需求方面的信息，估计新系统可能的开发和运行成本及其效益。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。

个别项目可能有例外，但一个典型的项目生命周期如图 1-3 所示，其中各阶段稍为详细的描述如下：

- **需求分析** 弄清用户对项目要实现的系统的详细需求。当项目被评价后，就可按需求确定地执行某些工作，但原来掌握的信息需要更

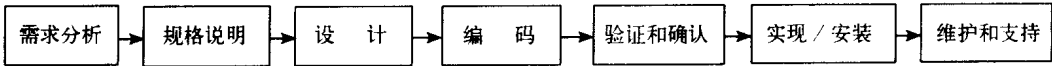


图 1-3 典型的项目生命周期

新和补充。要探索用户需求分析的各种方法。例如，可能要比拟是采用只满足用户的某些需要（但不是全部需要）的廉价小系统，还是采用价格高一些而功能较齐全的系统。

- **规格说明** 是所建议的系统要做什么的详细文档。
- **设计** 设计必须按照规格说明的要求来进行。前面已经讲到，设计分为两步：一是关心应用外观的外部设计或者说是用户界面设计；二是产生物理设计，它是将数据和软件过程进行内部结构化。
- **编码** 这可能涉及到用过程语言（如 C 和 Ada）来编写代码，或者可能涉及到使用像 Microsoft Access 这样的应用建造工具。即使软件不是从零开始编写，为了满足新的应用的需要，也可能要对基本的程序包进行一些修改。
- **验证和确认** 不管软件是否专为当前应用程序开发，都需要仔细地进行测试，以检查提出的系统是否满足其需求。
- **实现/安装** 一些系统开发人员把项目设计后的整个过程称之为“实现”（即设计的实现），而有些人坚持认为这个词是指在软件开发结束后对系统的安装。后者包括设置可运行的数据文件和系统参数、编写用户手册以及对新系统的用户培训。
- **维护和支持** 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，其工资单和其它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？

1.6 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，也可以设想将需要：

图 1-3 建议这些阶段必须严格按照顺序来执行。在第 4 章将看到可采用迭代等其它方法。然而，仍将进行这里列出的实际活动。

- 分析软件需求。
- 设计和编写测试用例，用以检查每个需求是否已经满足。
- 为每一个测试用例创建测试脚本和期望结果。
- 比较实际结果与期望结果，并标识它们不一致的地方。

通常方法与一类活动有关，计划使用该方法（或其它方法）并将其转化为一组真实活动，并标识出每一个活动：

- 开始日期和结束日期。
- 谁负责执行。
- 要使用什么工具和材料。

在当前这个语境中，“材料”可以是信息，如需求文档。

对于复杂的规程，可以使用顺序的或并行的几种方法。一个方法的输出可能是另一个方法的输入。一组方法或一组技术往往称之为方法学。例如，面向对象设计可以认为是由若干作为组成部分的方法构成的方法学。

见 Simon Bennett, Steve McRobb 和 Ray Farmer 的 *Object Oriented Systems Analysis and Design using UML*, McGraw-Hill, 1999。

1.7 软件项目的分类方法

区分软件项目的主要类型非常重要，因为在某一语境下适用的东西，可能在另一语境下就不适用。例如，有些人已经建议使用面向对象的方法，虽然在很多语境下是有用的，但可能在围绕关系数据库设计应用程序时并不理想。

1.7.1 信息系统与嵌入式系统

可能要对信息系统与嵌入式系统进行区别。概略地讲，两者的不同之处在于，前者的系统界面是与组织的接口，而后的系统界面是与机器的接口。前者如库存管理系统，是一个组织管理订购备料的信息系统。后者如嵌入式（或过程控制）系统，可以是一个在建筑物中控制空调设备的系统。有些系统可能兼有二者所具有的要素，例如上述库存管理系统也可以控制一个自动化仓库。

练习 1.3 计算机操作系统是信息系统还是嵌入式系统？

1.7.2 目的与产品

要区别项目的目标是为了生产一种产品，还是为了满足一定目的。

一个项目可能是生产一种其细节是由客户规定的产品。客户负责证实这个产品。

另一方面，项目可能是为了满足一定目的。可能有多种方法来达到这些目的。可以实现一个新的信息系统，来改进对组织的内部用户或外部用户所提供的服务。作为目标的服务标准将是协议的主题，而不是特定信息系统的特征。

嵌入式系统也叫实时系统或工业系统。

当组织外包功能给外部服务的供应商时，服务标准协议变得越来越重要了。

很多软件项目有两个阶段。第一阶段是目的驱动项目，这导致产生一个推荐的活动过程，甚至详细说明一个新软件系统，以满足所标识的需求。第二阶段是一个实际创建该软件产品的项目。

练习 1.4 在练习 1.2 中所描述的 Brightmouth 学院实现独立的工资发放系统的项目是目的驱动项目还是产品驱动项目？

1.8 什么是管理

开放大学软件项目管理模块（1987）建议表明，管理包括下列活动：

- 计划：决定要做什么。
- 组织：进行安排。
- 人员：选择合适的人员来完成作业等。
- 指导：作出指示。
- 监督：检查进展。
- 控制：采取行动以清除项目的障碍。
- 革新：提出新的解决方案。
- 代表：与用户沟通等。

获得开放大学资料的方便途径见 D. Ince, H. Sharp 和 M. Woodman 的 *Introduction to Software Project Management and Quality Assurance*, McGraw-Hill, 1993。

练习 1.5 Paul Duggan 是软件开发部经理。星期二上午 10:00，他和他的下属部门领导与集团经理开会讨论来年的人员需求。Paul 已经起草了一份叫“邀请”的关于人员的文件，这是根据他的部门明年的工作计划作出的。这份文件在会议上进行了讨论。下午 2:00，Paul 和他的高级职员会晤，讨论他的部门正在进行的一个重要项目。该项目有一名程序员出了车祸，将住院一段时间。为保证项目的进度，决定将另一组任务不紧迫的一名组员调到这个项目，以临时替代做些不很紧迫的工作，但这需要一周的时间来安排。Paul 一方面必须打电话给人事部门，告诉他们关于人员替补的问题，同时必须打电话给用户，解释项目可能延期的原因和谁在做这件不很紧迫的工作。

指出在这一天中 Paul 在不同点所作出的响应属于上面八项管理职责中的哪些职责。

1.9 软件项目的问题

决定“软件项目管理”应该覆盖什么内容的一种方法是，考虑在软件项目管理中应该关注的问题是什么。

传统上，管理是对组织内不同类事物的维护。当技术使得这些任务由更复杂的组织承担时，在该组织的内部，到处都有许多管理任务，有许多管理系统（而没有许多经理）。然而，成功的项目通常有一个对成功负责的人。这样的人专注于克服影响成功的各种障碍，是主要的故障检修员，他们的工

作可能由项目中遇到的问题来决定。Thayer 公布了一项对经理的调查, Pyster 和 Wood 指出了下列常见的经验问题:

- 较差的评估和计划。
- 缺乏质量标准和度量。
- 缺乏作出组织决策的指南。
- 缺乏使进展可见的技术。
- 角色定义(谁干什么)较差。
- 不正确的成功准则。

以上是从项目经理的角度看项目。从项目组成员的角度来看, 又是什么样的呢? 以下是刚完成一年生产实习的许多学生在计算与信息系统课上提出的问题:

- 工作的规格说明不合适。
- 管理部门忽视 IT。
- 缺乏应用领域知识。
- 缺乏标准。
- 文档缺乏更新。
- 前趋活动不能按时完成, 包括设备不能及时到位。
- 用户和技术人员之间缺乏交流。
- 缺乏交流导致工作重复。
- 缺乏承诺, 尤其是当项目全靠一个人, 然后这个人又离开时。
- 技术专长的范围狭窄。
- 改变法定的需求。
- 软件环境的改变。
- 截止期限的压力。
- 缺乏质量控制。
- 远程管理。
- 缺乏培训。

学生指出了很多由于缺乏交流而产生的问题。导致缺乏交流的一个因素是 IT 学科的范围很广: 一个组织可能由很多只精通某些软件技术和工具的人员所组成, 他们对他/她的同事所用的软件和工具则一无所知。因此交流问题也就显得突出了。

计算机项目产品的客户面临什么问题呢? 这是近来的一些新闻报道:

- 美国国内税收系统在花费 40 亿美元后放弃了税收系统现代化项目。
- 加州花费 10 亿美元在无用的福利数据库系统上。
- 英国耗资 3.39 亿英镑的空中交通管制系统已拖后进度两年。
- 贴现股票经纪业公司 50 个人工作 14 小时(一天多)来纠正三个月的办公室工作记录, 报告指出新系统未经充分测试就匆忙地投入使用。

有关该调查更详细的内容可参阅“软件工程项目管理中的主要问题”一文, 见 *IEEE Transactions on Software Engineering*, 7, 333-342, 1981。

Stephen Flower 的 *Software Failure, Management Failure*, Wiley, 1996, 是关于计算机项目失败的有趣的调查。