

21

21 世纪全国高校应用人才培养信息技术类规划教材



# 软件工程案例教程

毕硕本 卢桂香 编著



北京大学出版社  
PEKING UNIVERSITY PRESS

TP311.5/196

2007

21 世纪全国高校应用人才培养信息技术类规划教材

# 软件工程案例教程

毕硕本 卢桂香 编著



北京大学出版社  
PEKING UNIVERSITY PRESS

## 内 容 简 介

本书系统地阐述了软件工程这一领域内的基本概念、原理与方法,共包括8章,主要内容有:软件工程概述、可行性研究、需求分析与方法、总体设计与方法、详细设计与方法、工程实施、软件工程测试、软件工程维护等。

全书内容丰富、组织严谨,原理和方法结合密切,丰富的图表和应用实例便于读者自学。

本书可作为大专院校软件工程专业、计算机科学技术专业、计算机应用专业以及其他相关专业高年级本科生的教材,同时可供从事软件工程、计算机应用、计算机软件专业及其相关专业的科研人员、软件开发人员以及有关大专院校的师生参考。

## 图书在版编目(CIP)数据

软件工程案例教程/毕硕本,卢桂香编著. —北京:北京大学出版社,2007.8  
(21世纪全国高校应用人才培养信息技术类规划教材)  
ISBN 978-7-301-12505-2

I. 软… II. ①毕… ②卢… III. 软件工程—案例—高等学校—教材 IV. TP311.5

中国版本图书馆CIP数据核字(2007)第100188号

书 名: 软件工程案例教程

著作责任者: 毕硕本 卢桂香 编著

责任编辑: 温丹丹

标准书号: ISBN 978-7-301-12505-2/TP·0906

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路205号100871

电 话: 邮购部 62752015 发行部 62750672 编辑部 62765126 出版部 62754962

网 址: <http://www.pup.cn>

电子信箱: [xxjs@pup.pku.edu.cn](mailto:xxjs@pup.pku.edu.cn)

印 刷 者: 北京宏伟双华印刷有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787毫米×1092毫米 16开本 13.75印张 330千字

2007年8月第1版 2007年8月第1次印刷

定 价: 25.00元

---

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: 010-62752024; 电子信箱: [fd@pup.pku.edu.cn](mailto:fd@pup.pku.edu.cn)

# 前 言

作者从事软件工程开发 20 年，从事软件工程教学 5 年，根据作者的参考和使用的几十本软件工程教材来看，现有的这些教材大都侧重理论的传授，教材中的案例较少，而且没有利用一个完整、系统的软件工程案例进行教学活动。同时，由于本科生普遍缺乏软件工程项目开发的实践经验，因此学习软件工程课程感觉非常抽象、空泛与枯燥。为改变目前的软件工程课程教材的这种抽象、空泛的现状，作者编写了本教材。

本教材力图改变软件工程教学抽象、单调、死板的现状，利用精选的丰富的案例进行软件工程教学，而且利用若干个完整的案例贯穿软件工程开发活动的整个过程来进行软件工程思想的传授，使得软件工程教材不再是难以理解和缺乏生动形象的教材，以调动学生的形象思维和学习兴趣，取得良好的教学效果。

本教材共包括 8 章。第 1 章概述了软件工程领域的基本内容，第 2 章介绍了软件工程的可行性研究内容，第 3 章介绍了软件工程的需求分析内容，以及软件工程的需求分析所涉及的几种分析方法。

第 4 章进入了软件工程的设计阶段，该章介绍软件工程总体设计包含的基本内容，是软件工程的基础和关键。第 5 章介绍了软件工程的详细设计所包含的内容以及软件工程的设计阶段所涉及的设计方法。

第 6、7 章进入软件工程阶段的实施与测试阶段。实施阶段主要介绍程序编制有关的内容，测试阶段主要介绍测试的要求、软件测试的方法、测试用例的设计以及软件调试与试运行的有关内容。最后，第 8 章介绍了软件维护涉及的主要内容。

本教材定位在软件工程思想的传统方法和面向对象技术方法两个方面上，内容编排方法多样、新颖，案例生动、形象而且贴近学生的生活，易于被学生接受、理解。采用案例形式来描述软件开发的全过程，案例贯穿全书始终，通过案例说明，帮助读者消化和理解所学内容，迅速提高实践能力。本书内容与结构体系完整、严密、紧凑，写作风格严谨、流畅，易读性强，是作者多年来软件工程教学中认真思考与努力实践的成果总结。

本教材是作者在多年从事该领域的工作，并参阅了大量国内外有关论著、期刊文献，以及与相关专家、学者交流的基础上编写成的。本教材在编写和修改过程中，得到了许多同志的关心和帮助，特别是南京信息工程大学计算机与软件学院副院长傅德胜教授、顾韵华副教授给予了热忱的鼓励，提出了许多宝贵意见和建议，在此表示衷心感谢。

本书可作为大学院校软件工程专业、计算机科学技术专业、计算机应用专业以及其他相关专业高年级本科生的教材，同时可供从事软件工程、计算机应用、计算机软件专业及其相关专业的科研人员、软件开发人员以及有关大专院校的师生参考。

由于作者水平有限，加上时间仓促，难免有错误和不妥之处，恳请读者批评指正。

编 者  
2007 年 6 月

# 目 录

第1章 软件工程概述.....	1
1.1 软件工程的概​​念.....	1
1.1.1 软件的主要特点.....	1
1.1.2 软件开发过程中的问题.....	1
1.1.3 原因分析.....	2
1.1.4 软件工程的概​​念.....	3
1.2 软件工程的各个阶段.....	5
1.2.1 前期工程.....	6
1.2.2 设计工程.....	7
1.2.3 数据工程.....	7
1.2.4 工程实施.....	8
1.2.5 维护工程.....	8
1.3 软件生存周期模型.....	9
1.3.1 概述.....	9
1.3.2 瀑布模型.....	9
1.3.3 增量模型.....	11
1.3.4 螺旋模型.....	11
1.3.5 喷泉模型.....	12
1.4 习题.....	12
第2章 软件工程的可行性研究.....	13
2.1 软件工程可行性研究案例.....	13
2.2 软件工程调研.....	21
2.2.1 现状调查.....	21
2.2.2 需要调查的内容.....	22
2.2.3 调查内容的组织和分析.....	23
2.3 可行性研究的基本内容.....	25
2.3.1 可行性研究的目的.....	25
2.3.2 可行性研究的任务.....	26
2.3.3 可行性研究的步骤.....	27
2.4 成本-效益分析.....	28
2.4.1 几种度量效益的方法.....	28
2.4.2 成本-效益的分析.....	29
2.5 项目开发计划.....	31
2.5.1 制订项目开发计划的原则.....	31

2.5.2	制订项目开发计划的方法 .....	32
2.5.3	推算各阶段时间的方法 .....	33
2.6	习题 .....	34
<b>第3章</b>	<b>软件工程的需求分析 .....</b>	<b>35</b>
3.1	软件工程需求分析案例 .....	35
3.2	需求分析的基本内容 .....	45
3.2.1	需求分析的必要性 .....	46
3.2.2	需求分析的原则 .....	46
3.2.3	用户需求的类型 .....	47
3.2.4	需求分析的方法 .....	48
3.2.5	数据流程图 .....	48
3.2.6	数据字典 .....	51
3.3	结构化分析方法 .....	53
3.3.1	概述 .....	53
3.3.2	结构化分析方法 .....	54
3.4	面向对象的分析方法 .....	56
3.4.1	面向对象方法简介 .....	56
3.4.2	面向对象分析概述 .....	57
3.4.3	论域分析 .....	58
3.4.4	应用分析 .....	60
3.5	习题 .....	61
<b>第4章</b>	<b>软件工程的总体设计 .....</b>	<b>62</b>
4.1	软件工程总体设计案例 .....	62
4.2	总体设计概述 .....	71
4.2.1	系统设计在软件开发中的位置 .....	72
4.2.2	总体设计过程中需要完成的工作 .....	73
4.3	总体设计的基本内容 .....	75
4.3.1	概述 .....	75
4.3.2	总体设计的目标 .....	75
4.3.3	总体设计的步骤 .....	75
4.3.4	总体设计的基本任务 .....	75
4.4	软件结构设计 .....	77
4.4.1	概述 .....	77
4.4.2	结构化设计的优点 .....	77
4.4.3	开发和维护软件的费用 .....	78
4.4.4	结构化设计的适用范围 .....	78
4.4.5	结构化分析 (SA) 和结构化设计 (SD) 的关系 .....	79
4.4.6	结构设计的基本概念 .....	80
4.4.7	模块的独立性 .....	81
4.4.8	软件结构图 .....	85

4.4.9	软件结构形态.....	86
4.4.10	影响范围和控制范围.....	87
4.4.11	软件结构设计准则.....	89
4.5	数据库设计.....	90
4.5.1	数据库分析的过程.....	91
4.5.2	数据的管理模式.....	92
4.5.3	数据库设计的目标.....	93
4.5.4	数据库设计的主要内容.....	93
4.5.5	数据库设计的步骤.....	93
4.5.6	数据库的逻辑设计.....	95
4.5.7	数据库的物理设计.....	96
4.6	习题.....	97
第5章	软件工程的详细设计.....	99
5.1	软件工程详细设计案例.....	99
5.2	详细设计的基本内容.....	109
5.2.1	概述.....	109
5.2.2	详细设计的基本任务.....	110
5.2.3	详细设计方法.....	111
5.2.4	程序流程图.....	112
5.2.5	N-S图.....	115
5.2.6	PAD图.....	116
5.2.7	判定表.....	118
5.2.8	PDL语言.....	119
5.3	用户界面设计.....	120
5.3.1	用户界面应具备的特性.....	120
5.3.2	用户界面设计的任务分析.....	122
5.3.3	用户界面的任务和工作设计.....	124
5.3.4	界面设计的基本类型.....	125
5.3.5	界面设计原则.....	126
5.3.6	软件界面设计中的要素.....	127
5.3.7	软件的基本界面样式.....	129
5.3.8	软件的界面设计方法.....	133
5.4	结构化设计方法.....	134
5.4.1	系统结构图的基本组成.....	134
5.4.2	数据流图变换为系统结构图的方法.....	135
5.4.3	结构化设计的审查.....	138
5.5	面向对象设计概述.....	139
5.6	面向对象的UML方法.....	140
5.6.1	UML概述.....	140
5.6.2	通用模型元素.....	143

5.6.3	用例模型	144
5.6.4	静态模型	147
5.6.5	动态模型	153
5.6.6	实现模型	161
5.7	习题	164
<b>第6章</b>	<b>软件工程的实施</b>	<b>166</b>
6.1	软件工程编程案例	166
6.2	概述	167
6.2.1	实施阶段的准备工作	167
6.2.2	硬软件引进	168
6.2.3	软件工程对编程的技术要求	169
6.2.4	软件工程对编程的质量要求	169
6.3	程序编制	170
6.3.1	程序编制的一般要求	170
6.3.2	编程语言的特征及选择	171
6.3.3	程序设计风格	173
6.4	习题	174
<b>第7章</b>	<b>软件工程的测试</b>	<b>175</b>
7.1	软件工程技术案例	175
7.2	软件测试	175
7.2.1	概述	175
7.2.2	软件测试的要求	176
7.2.3	软件测试过程	179
7.2.4	单元测试	181
7.2.5	集成测试	182
7.2.6	确认测试	185
7.2.7	软件测试的方法	186
7.3	测试用例的设计	187
7.3.1	测试用例的设计原则	187
7.3.2	黑盒测试方法用例的设计	187
7.3.3	白盒测试方法用例的设计	191
7.4	软件调试	196
7.4.1	调试的目的	196
7.4.2	调试的步骤	196
7.4.3	调试的原则	197
7.4.4	调试方法	198
7.5	习题	201
<b>第8章</b>	<b>软件维护工程</b>	<b>202</b>
8.1	软件维护案例	202
8.2	概述	202



---

8.2.1 软件维护的类型.....	202
8.2.2 软件维护的困难.....	203
8.2.3 软件维护的费用.....	203
8.2.4 软件维护的分类.....	203
8.3 软件系统的维护.....	204
8.3.1 概述.....	204
8.3.2 软件维护的过程.....	205
8.3.3 软件维护技术.....	205
8.3.4 影响维护工作量的因素.....	206
8.3.5 软件维护的策略.....	207
8.3.6 维护成本.....	207
8.4 习题.....	208
参考文献.....	209

# 第 1 章 软件工程概述

本章概述软件工程的基本内容，首先介绍软件开发过程中存在的问题，然后简要介绍软件工程的生存周期，最后介绍软件生存周期模型。

## 1.1 软件工程的观念

软件工程具有一般工程的共同特性，又有自己的独特个性。这里首先分析软件的主要特点，找出软件开发中存在的问题，并进行原因分析，最后给出软件工程的观念。

### 1.1.1 软件的主要特点

#### 1. 几个基本概念

软件是计算机系统中与硬件相互依存的另一部分，它是包括程序，数据及其相关文档的完整集合。

程序是按事先设计的功能和性能要求执行的指令序列（Instructions）。

数据是使程序能正常操纵信息的数据结构（Data Structures）。

文档是与程序开发，维护和使用有关的图文材料（Documents）。

#### 2. 软件的主要特点

一般而言，软件的开发和运行常受到计算机系统的限制，对计算机系统有着不同程度的依赖性。从本质上看，软件本身是复杂的，主要表现在程序逻辑结构上的复杂性。从应用要求上看，软件要解决的实际问题也是复杂的，软件成本相当昂贵。特别是，软件的开发至今尚未完全摆脱手工艺的开发方式，相当多的软件工作涉及到社会因素。所以要高效率地开发一个高质量的软件，需要从软件的特点入手。

软件一般有以下特点。

(1) 从表现形式看，软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。

(2) 从生产方式看，软件的生产与硬件不同，在它的开发过程中没有明显的制造过程，大多数软件仍是定制的。

(3) 从维护机制上看，在软件的运行和使用期间，没有硬件那样的机械磨损、老化问题。

(4) 从使用要求上看，一般情况下，软件产品不允许有误差。

### 1.1.2 软件开发过程中的问题

从软件的主要特点可知，软件开发过程中有许多需要解决的棘手问题，如软件费用、用

户需求、软件可维护性、软件可靠性、数据量、软件生产率和软件重用等。

(1) 经费预算经常突破, 完成时间一再拖延。由于缺乏软件开发的经验和软件开发数据的积累, 使得软件开发计划很难制订。主观盲目制订的计划, 执行起来和实际情况有很大差距, 使得开发经费一再突破。由于对工作量和开发难度估计不足, 进度计划无法按时完成, 开发时间一再拖延。

(2) 开发的软件不能满足用户的要求。开发初期对用户的要求了解不够明确, 未能得到明确表达。开发工作开始后, 软件人员和用户未能及时交换意见, 使得一些问题不能及时解决, 导致开发软件不能满足用户的要求, 使得开发工作失败。

(3) 开发的软件可维护性差。软件开发人员按各自的风格工作, 各行其是。程序结构不好, 运行维护时发现错误也很难修改, 导致软件维护性差。统计数据表明, 软件的维护费用占整个软件系统费用的 2/3, 而软件开发费用只占 1/3。软件维护之所以有如此大的花费, 是因为已经运行的软件还需排除隐含的错误, 新增加的功能要加入进去, 维护工作又是非常困难的, 效率是非常低的。因此, 如何提高软件的可维护性, 减少软件维护的工作量, 也是软件工程面临的主要问题之一。

(4) 开发的软件可靠性差。在软件开发过程中, 若在测试时, 没有严格的、完整的测试, 那么提交给用户的软件质量差, 在运行中就会暴露出大量的问题。这种不可靠的软件, 经常会影响系统正常工作, 甚至会发生事故, 造成生命财产的重大损失。在软件开发中, 通常会花费 40% 的代价进行测试和排错, 即使这样也不能保证以后不再发生错误, 为了提高软件可靠性, 就要付出足够的代价。

(5) 部分应用领域的软件开发所需数据工程量特别大, 特别是数据采集工作量十分大, 例如, 地理信息系统 (GIS) 应用软件。而由数据工程又带来一系列问题, 如可靠性差、周期延长、费用增长等。

(6) 软件需求与软件生产的矛盾日益加剧, 突出表现在软件生产率低。软件是知识高度密集的技术的综合产物, 计算机的广泛应用使得软件的需求量大幅度上升, 软件的发展远远不能适应社会对其迅速增长的要求, 而软件的生产又处于手工开发的状态, 软件生产率低, 使得各国都感到软件开发人员的不足。所以, 如何提高软件生产率, 是软件工程的又一重要问题。

(7) 软件可重用性差。提高软件的可重用性, 对于提高软件生产率、降低软件成本有着重要意义。当前的软件开发存在着大量的、重复的劳动, 耗费了不少的人力资源。软件的可重用有各种级别, 软件规格说明、软件模块、软件代码、软件文档等都可以是软件重用的单位。软件重用是软件工程中的一个重要研究课题, 软件重用的理论和技术至今尚未彻底解决。

### 1.1.3 原因分析

软件之所以存在以上问题, 是由软件开发过程中各方面的原因引起的, 必须引起高度的重视, 需要深入分析产生问题的根源。下面对存在的以上问题进行综合分析。

(1) 软件的规模越来越大, 结构越来越复杂。随着计算机应用的日益广泛, 需要开发的软件规模日益庞大, 软件结构也日益复杂。

[例 1-1] 1968 年美国航空公司订票系统达到 30 万条指令。

[例 1-2] IBM360 OS 第 16 版达到 100 万条指令, 花了 5 000 个人年。

[例 1-3] 1973 年美国阿波罗计划达到 1 千万条指令。

这些庞大软件的功能非常复杂,体现在处理功能的多样性和运行环境的多样性。有人曾估计,软件设计与硬件设计相比,其逻辑量要多达 10~100 倍。对于这种庞大规模的软件,其调用关系、接口信息复杂,数据结构也复杂,这种复杂程度超过了人所能接受的程度。

(2) 软件开发的管理困难。由于软件规模大,结构复杂,又具有无形性,导致管理困难,进度控制困难,质量控制困难,可靠性无法保证。

(3) 软件开发费用不断增加。软件生产是一种智力劳动,它是资金密集、人力密集的产业,大型软件投入人力多,周期长,费用上升很快。

(4) 软件开发技术落后。在 20 世纪 60 年代,人们注重一些计算机理论问题的研究,如编译原理、操作系统原理、数据库原理、人工智能原理、形式语言理论等,不注重软件开发技术的研究,用户要求的软件复杂性与软件技术解决复杂性的能力不相适应,它们之间的差距越来越大。

(5) 生产方式落后。软件仍然采用个体手工方式开发,根据个人习惯爱好工作,无章可循,无规范可依据,靠言传身教方式工作,缺乏工程化思想方法的指导。

(6) 开发工具落后,生产率提高缓慢。软件开发工具过于原始,没有出现高效率的开发工具,因而软件生产率低下。

[例 1-4] 在 1960~1980 年期间,计算机硬件的生产由于采用计算机辅助设计、自动生产线等先进工具,使硬件生产率提高了 100 万倍,而软件生产率只提高了 2 倍,相差十分悬殊。

(7) 软件处在快速上升分化发展过程中,更新变化很快,不可避免地存在不少发展过程中的衔接问题。

#### 1.1.4 软件工程的观念

[例 1-5] 美国 IBM 公司在 1963~1966 年开发的 IBM 360 机的操作系统。这一项目花了 5000 人一年的工作量,最多时有 1000 人投入开发工作,写出了近 100 万行源程序。据统计,这个操作系统每次发行的新版本都是从前一版本中找出 1000 个程序错误而修正的结果。

这个项目的负责人 F. D. Brooks 事后在总结了他在组织开发过程中的沉痛教训时说:“……正像一只逃亡的野兽落到泥潭中做垂死的挣扎,越是挣扎,陷得越深,最后无法逃脱灭顶的灾难……程序设计工作正像这样一个泥潭……一批批程序员被迫在泥潭中拼命挣扎……谁也没有料到问题竟会陷入这样的困境……”。IBM 360 操作系统的历史教训成为软件开发项目的典型事例为人们所记取。

(1) 软件工程的定义。为了克服类似例 1-5 一样的软件危机,人们从其他产业的工程化生产得到启示,于是在 1968 年北大西洋公约组织的工作会议上首先提出“软件工程”的概念,提出要用工程化的思想来开发软件。从此,软件生产进入了软件工程的时代。

Boehm 曾经为软件工程下了定义:“运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料”。这里对“设计”一词应有广义的理解,它应包括软件的要求分析和对软件进行修改时所进行的再设计活动。

1983 年 IEEE 给出的定义为:“软件工程是开发、运行、维护和修复软件的系统方法”,其中,“软件”的定义为:计算机程序、方法、规则、相关的文档资料,以及在计算机上运行时所必需的数据。

软件工程是用科学知识和技术原理来定义、开发、维护软件的一门学科。该定义说明了软件工程是计算机科学中的一个分支，其主要思想是在软件生产中用工程化的方法代替传统手工方法。工程化的方法借用了传统的工程设计原理的基本思想，采用了若干科学的、现代化的方法技术来开发软件。这种工程化的思想贯穿到需求分析、设计、实现，直到维护的整个过程。

(2) 软件工程的性质。软件工程是涉及计算机科学、工程科学、数学等领域的一门综合性的交叉学科。计算机科学中的研究成果均可用于软件工程，但计算机科学着重于原理和理论，而软件工程着重于如何建造一个软件系统。

软件工程要用工程科学中的观点来进行费用估算、制订进度、制订计划和方案；要用管理科学中的方法和原理进行软件生产的管理；要用数学的方法建立软件开发中的各种模型和各种算法，如可靠性模型，说明用户需求的形式化模型等。

(3) 软件工程的目標。软件工程是一门工程性学科，目的是从技术和管理上采取了多项措施，组织实施软件工程项目来建造一个大型软件系统，并最终希望得到项目的成功。所谓成功，是要达到以下几个目标：付出较低的开发成本，达到要求的软件功能，取得较好的软件性能，开发的软件易于移植，需要较低的维护费用，能按时完成开发任务、及时交付使用，开发的软件可靠性高。

在具体项目的实际开发中，企图让以上几个目标都达到理想的程度往往是非常困难的，而且上述目标很可能是互相冲突的。例如，若只顾降低开发成本，很可能同时也降低了软件的可靠性。另一方面，如果过于追求提高软件的性能，可能造成开发出的软件对硬件有较大的依赖，从而直接影响到软件的可移植性。

图 1-1 表明了软件工程目标之间存在的相互关系。其中有些目标之间是互补关系，例如，易于维护和高可靠性之间，低开发成本与按时交付之间。还有一些目标是彼此互斥的，例如，上述指出的互相冲突的情况。

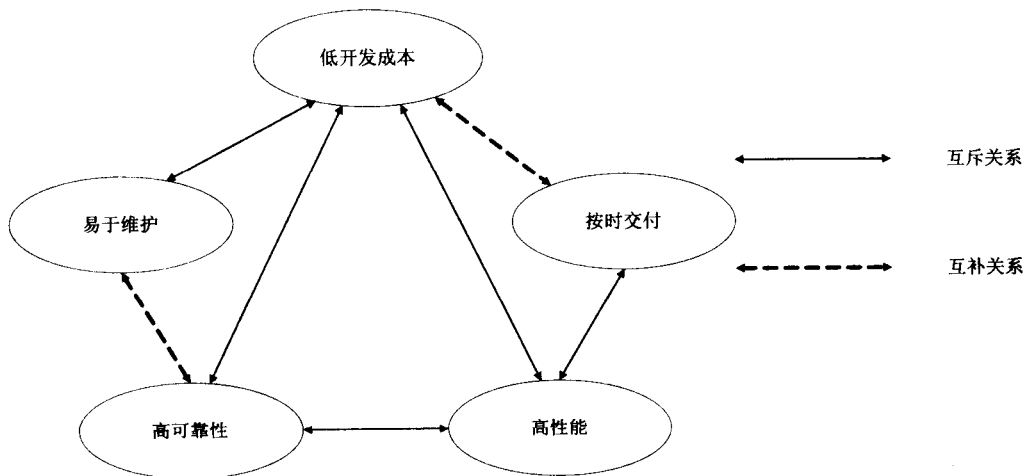


图 1-1 软件工程目标之间的关系

这里提到的几个目标很自然地成为判断软件开发方法或管理方法优劣的衡量尺度。实际上，实施软件开发项目就是力图在以上目标的冲突中取得一定程度的平衡。

(4) 软件工程的内容。软件工程研究的主要内容是指软件开发技术和软件开发管理两个方面。在软件开发技术中,它主要研究软件开发方法、软件开发过程、软件开发工具和环境。在软件开发管理中,它主要是研究软件管理学、软件经济学和软件心理学。

软件工程包括三个要素:方法、工具和过程。

软件工程方法为软件开发提供了“如何做”的技术。它包括了多方面的任务,如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法过程的设计、编码、测试以及维护等。软件工程方法常采用某一种特殊的语言或图形的表达方法及一套质量保证标准。

软件工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前,已经推出了许多软件工具,已经能够支持上述的软件工程方法。特别地,已经有人把诸多的软件工具集成起来,使得一种工具产生的信息可以为其他的工具所使用,这样建立起一种被称之为计算机辅助软件工程(CASE)的软件开发支撑系统。CASE将各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境。

软件工程的过程则是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理,以及软件开发各个阶段完成的里程碑。

软件工程就是包含上述方法、工具及过程在内的一些步骤。

软件工程的内容按照软件开发过程的先后顺序,包括前期工程、设计工程、数据工程、工程实施和维护工程5个阶段,每个阶段都以工程化原理作为指导,以工程化方法作为手段,并以质量控制、工程标准和工程管理作为保障,确保软件的开发成功。

## 1.2 软件工程的各个阶段

软件工程的生存周期是指一个软件从提出开发要求开始直到该软件报废为止的整个时期。把整个生存周期划分为若干阶段,使得每个阶段有明确的任务,把规模大、结构复杂和管理复杂的软件开发变得容易控制和管理。

软件的开发不仅有其既定的目标,而且有其阶段性。软件工程整个生存周期划分为4个阶段,一般而言,包括以下4个阶段,系统分析(也称需求分析)、系统设计(包括总体设计与详细设计)、工程实施(包括编程与测试等)、软件维护。

软件生存周期的各个阶段有不同的划分。软件规模、种类、开发环境,以及开发使用的方法都能影响软件生存周期的划分。在划分软件生存周期阶段时,应遵循的基本原则是各阶段的任务应尽可能的独立,同一阶段各项任务的性质尽可能相同,从而降低每个阶段任务的复杂程度,简化不同阶段之间的联系,有利于软件工程项目的开发与管理。通常,软件生存周期包括工程调研、可行性研究、需求分析、总体设计、详细设计、程序编制、测试、调试、试运行与维护等活动,可将这些活动以适当方式分配到不同工程阶段去完成。

软件工程过程没有规定一个特定的生存周期模型或软件开发方法,各软件开发机构可为其开发项目选择一种生存周期模型,并将软件工程过程所含的过程、活动和任务影射到该模型中,也可以选择和使用软件开发方法来执行适合于其软件工程项目的活动和任务。

按照工程化阶段的思想进行划分，软件工程生存周期可划分为包括：前期工程、设计工程、数据工程、工程实施、维护工程 5 个阶段。本书正是基于以上 5 个工程阶段进行材料的编写和组织的，以下简要介绍各个工程阶段的情况。

### 1.2.1 前期工程

软件前期工程阶段，包括工程调研、可行性研究、需求分析等子阶段。

#### 1. 工程调研

根据软件工程开发的基本目标和技术要求，对国内外相关软件工程项目通过走访、交谈、函件往来、资料检索等方式进行调研，确定该领域的软件的发展现状与存在问题，从而为拟开展的软件工程项目提供有价值的参考资料。

#### 2. 可行性研究

可行性研究阶段必须要回答的问题是“要解决的问题是什么？”“该问题有行得通的解决办法吗？”“若有解决问题的办法，则需要多少费用？”“需要多少资源？”“需要多少时间？”要回答这些问题，就要进行问题定义、可行性研究，并制订软件工程项目开发计划。

用户提出一个软件开发要求后，系统分析员首先要解决该软件工程项目的性质是什么，它是数据处理问题还是实时控制问题，它是科学计算问题还是人工智能问题等。还要明确该问题的目标是什么，该工程项目的规模如何等。通过系统分析员对用户和使用部门负责人访问和调查、开会讨论，就可解决这些问题。

在清楚了问题的性质、目标、规模后，还要确定问题有没有行得通的解决办法。系统分析员要进行压缩和简化的需求分析和设计，也就是在高层次上进行分析和设计，探索这个问题是否值得去解决，是否有可行的解决办法。

经过可行性研究后，确定该问题值得去解决，然后制订软件工程项目开发计划。根据开发项目的目标、功能、性质及规模，估计项目需要的资源，即需要的计算机硬件资源、软件开发工具和应用软件包，需要的开发人员数目及层次。还要对软件开发费用做出估算，对开发进度做出估计，然后制订完成任务的实施计划。最后，将包含软件工程项目开发计划的可行性研究报告提交给管理部门与软件系统用户进行审查。

#### 3. 需求分析

需求分析阶段的任务不是具体地解决问题，而是准确地确定“软件系统必须做什么”，软件系统必须具备哪些功能。

用户了解他们所面对的问题，知道必须做什么，但是通常不能完整、准确地表达出来，也不知道怎样用计算机解决他们的问题。而软件开发人员虽然知道怎样用软件完成人们提出的各种功能要求，但是对用户的具体业务和需求不完全清楚，这是需求分析阶段的困难所在。

系统分析员要和用户密切配合，充分交流各自的理解，充分理解用户的业务流程，完整、全面地收集、分析用户业务中的信息和所做的处理，从中分析出用户要求的功能和性能，并完整、准确地表达出来。这一阶段要根据需求分析所做的工作编制出软件需求说明书。

## 1.2.2 设计工程

软件设计工程阶段,包括总体设计、数据库设计、应用模型设计、详细设计等子阶段。

(1) 总体设计。在总体设计阶段,软件开发人员把确定的各项软件功能需求转换成需要的软件体系结构,在该体系结构中,每个成分都是意义明确的模块,即每个模块都和某些功能需求相对应。因此,总体设计就是设计软件的结构,确定该结构由哪些模块组成,这些模块的层次结构是怎样的,这些模块的调用关系是怎样的,每个模块的功能是什么。同时还要设计该软件工程项目应用系统的数据结构和数据库结构,即应用系统要存储什么样的数据,这些数据是什么样的结构,它们之间有什么样的关系等。

(2) 数据库设计。数据库是一个信息系统的基本且重要的组成部分。在一个软件系统的运行过程中,数据库发挥着核心的作用。数据库设计是指对于一个给定的应用环境,提供一个确定的最优数据模型与处理模式的逻辑设计,以及一个确定数据存储结构与存取方法的物理设计,建立能反映现实世界信息和信息联系,满足用户要求,又能被某个数据库管理系统(DBMS)所接受,同时能实现软件系统目标并有效存取数据的数据库。

(3) 应用模型设计。由于各种应用软件具有交叉性、综合性的特点,所以软件的应用领域相当广泛,而每种类型的应用软件都有自己独特的特点,这个特点主要体现在应用领域模型的构建和实现上。所以在进行应用型软件工程项目开发的分析阶段,一项重要的工作是针对拟开发领域的特殊技术要求,运用并分析该领域的应用模型,设计与实现该应用模型的技术方法。

(4) 详细设计。详细设计阶段就是为每个软件模块完成的功能进行具体的描述,并把功能描述转变为精确的、结构化的过程描述。即该模块的控制结构是怎样的,先做什么,后做什么,有什么样的条件判定,有些什么重要处理等,并用相应的表示工具把这些控制结构表示出来。

## 1.2.3 数据工程

数据工程阶段,包括数据预处理、数据采集、数据处理等子阶段。

### 1. 数据预处理

在数据工程阶段,开发软件的第一步就是将应用领域中现有的各种类型的实体数据经过一定的规范化前期处理,根据应用软件数据处理的需要,变换成一致的数据格式,一致的数据量纲等,该项工作称为数据预处理。

[例 1-6] 在 GIS 应用软件的开发过程中,无论用何种方法获取的原始数据,都可能存在着数字化过程中不可避免的引入错误,数字化数据与使用格式不一致,各种数据来源的比例尺和投影不统一,各幅地图数据之间的不匹配,以及地图比例尺之间、地图比例尺与数字化仪的长度单位之间不一致等,因此必须通过数据预处理,才能获得净化的数据文件,使采集的数据符合规范化标准。

### 2. 数据采集

在数据工程阶段,开发软件的重要一环就是通过一定的技术手段获取到应用领域实体的各种类型的数据,并建立为数据文件或输入到数据库中,该项工作就是数据采集。



[例 1-7] GIS 应用软件的数据采集需进行 4 方面的工作, 即野外数据的采集、图形数据的采集、属性数据的采集和图形数据与属性数据的连接。野外数据的采集主要有三种: GPS 方法、摄影测量方法和测量法。地图数据是 GIS 主要的信息源, 对它的采集主要是指对地图进行数字化, 主要有两种方式, 即手扶跟踪数字化和扫描数字化。

### 3. 数据处理

在数据工程阶段, 开发软件的最后一步就是将应用领域中经过数据预处理后所采集的数据, 进行一定的变换处理, 产生软件运算所需要的数据, 该项工作称为数据预处理。

[例 1-8] 在 GIS 应用软件的开发过程中, 数据处理工作包括格式转换、图形单元的修改与增删、图幅拼接、坐标转换、几何纠正、数据编辑处理、拓扑结构及拓扑关系的自动生成等数据库建立前的工作。

## 1.2.4 工程实施

软件工程实施阶段, 包括程序编制、测试、调试、试运行等子阶段。

(1) 程序编制。程序编制阶段就是把每个模块的控制结构转换成计算机可接受的程序代码, 即写成以某特定程序设计语言表示的“源程序清单”。编写出的程序应是结构好, 清晰易读, 并且与设计相一致。

(2) 测试。测试是保证软件质量的重要手段, 其主要方式是在设计测试用例的基础上检验软件的各个组成部分。测试分为模块测试、组装测试、确认测试。模块测试是查找各模块在功能和结构上存在的问题。组装测试是将各模块按一定的顺序组装起来进行测试, 主要是查找各模块之间接口上存在的问题。确认测试是按软件需求说明书上的功能逐项进行的, 发现不满足用户需求的问题, 决定开发的软件是否合格、能否交付用户使用等。

(3) 调试。软件测试的目的是尽可能多地发现程序中的错误, 而调试则是在进行了成功的测试之后才开始的工作。调试的目的是确定错误的原因和位置, 并改正错误。调试是通过现象, 找出原因的一个思维分析的过程。调试方法包括简单的调试方法、归纳法、演绎法、回溯法 4 种。

(4) 试运行。试运行是保证最终交付给用户的软件质量的重要手段, 软件试运行应由软件开发人员和用户共同进行。在试运行过程中要进行正确性完善和适应性完善。试运行的时间应视系统的规模和复杂程度而定, 一般为 1~3 个月。

## 1.2.5 维护工程

软件维护工程阶段, 包括数据库维护、软硬件维护等方面。

(1) 数据库维护。软件需要的数据随着应用规模的日益扩大而迅速变化, 所有软件中的数据均需要经常地进行维护和更新。所以应根据软件系统的规模和实际需求, 建立数据维护更新机制, 规定数据维护更新的周期, 使软件系统的所有数据均相对地始终处于最新的状态。

(2) 软件维护和硬件维护。软件维护是软件生存周期中时间最长的阶段。已交付的软件投入正式使用后, 便进入软件维护阶段, 它可以持续几年甚至几十年。软件运行过程中可能由于各方面的原因, 需要对它进行修改。其原因可能是运行中发现了软件隐含的错误而需要修改, 也可能是为了适应变化了的软件工作环境而需要做适当变更, 也可能是因为用户业务