



“十二五”普通高等教育本科国家级规划教材



普通高等教育“十一五”国家级规划教材



普通高等教育计算机规划教材

软件开发技术基础

第 3 版

赵英良 主 编
仇国巍 卫颜俊 李尊朝 等编著



提供电子教案和习题解答

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS





“十二五”普通高等教育本科国家级规划教材



普通高等教育“十一五”国家级规划教材

普通高等教育计算机规划教材

软件开发技术基础

第3版

赵英良 主编

仇国巍 卫颜俊 李尊朝 等编著

冯博琴 审



机械工业出版社

本书根据教育部高等学校计算机基础课程教学指导委员会《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》中软件类课程的要求,结合非计算机专业软件开发的特点和一般软件开发必备技术组织编写。

本书介绍了软件开发中常用的基本原理、方法和技术。本书主要内容包括软件工程基础、数据结构及其应用、操作系统及相关程序设计、数据库及应用程序开发、网络软件开发技术、多媒体编程技术。附录 A 为 Visual C++6.0 Windows 编程基础;附录 B 为实验。本书以 C/C++ 为编程语言,每章均有一定数量的例题和习题,旨在锻炼学生的软件开发能力。

本书结构合理,条理清晰,内容实用,可作为普通高等院校理工类非计算机专业的软件技术基础课程教材,也可供相关技术人员参考。

本书配有电子教案,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ: 2966938356, 电话: 010 - 88379739)。

图书在版编目(CIP)数据

软件开发技术基础 / 赵英良主编. —3 版. —北京: 机械工业出版社, 2015. 7

“十二五”普通高等教育本科国家级规划教材 普通高等教育“十一五”
国家级规划教材 普通高等教育计算机规划教材
ISBN 978-7-111-50970-7

I. ①软… II. ①赵… III. ①软件开发 - 高等学校 - 教材
IV. ①TP311.52

中国版本图书馆 CIP 数据核字 (2015) 第 170858 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 和庆娣 责任编辑: 和庆娣 张 恒

责任校对: 张艳霞 责任印制: 乔 宇

北京铭成印刷有限公司印刷

2015 年 10 月第 3 版 · 第 1 次印刷

184mm × 260mm · 19 印张 · 471 千字

0001 - 3000 册

标准书号: ISBN 978-7-111-50970-7

定价: 45.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88379833

机工官网: www.cmpbook.com

读者购书热线: (010) 88379649

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

封面无防伪标均为盗版

金书网: www.golden-book.com

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

希望计算机教育界的专家和老能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

前言

程序设计是软件开发的基础课程。然而，虽是学习了程序设计，但要进行真正的软件开发，似乎是路漫漫其修远兮。即使编写一个很小的应用软件，也会或多或少地用到数据库、网络、多媒体等技术，数据结构和操作系统的功能调用更是核心和基础，然而这些技术在一般的程序设计教材中是很少全面涉及的。本书为程序设计过渡到软件开发提供技术支持。本书前两版受到读者欢迎，第2版为普通高等教育“十一五”国家级规划教材，2012年又入选“十二五”普通高等教育本科国家级规划教材，西安交通大学的“软件开发技术基础”课程2014年升级为国家级精品资源共享课（网址：http://www.icourse.cn/course_static/courseo-4126.html）。应读者建议，本书进行了如下修改：

第1章对传统方法学的细节进行了进一步细化，使得内容更实用，如给出软件详细设计的工具和模块的划分方法，软件测试方法给出了实例等。

第2章的主要修改是将原来基于类的例题改成了基于结构体的例题，使得学过C语言或C++语言的读者都比较容易理解。

第3章调整了结构，将操作系统管理的理论知识放在第3.2节中讲述，相关的程序设计技术放在第3.3节讲述。

第4章补充了在Windows 7下建立Access数据源的方法，方便读者实践。

第5章调整了结构，简化了内容，特别是对例题进行了精简，使条理更加清晰，重点更加突出。

第6章调整了结构，删除了不必要的知识罗列，介绍了与多媒体有关的系统高级函数、MCI接口、Windows GDI和ActiveX控件的使用方法。

第3版的结构更加合理，条理更加清晰，内容更加实用。当然仍会有疏漏之处，希望读者提出更好的建议，也希望专家不吝赐教。

本书由赵英良主编，并改编了第1章和第6章，仇国巍改编了第2章和第5章，卫颜俊改编了第3章，李尊朝改编了第4章，参与改编的还有刘志强、薛涛、崔舒宁。冯博琴教授对本书进行了审阅并提出了建议。本书得到西安交通大学计算机教学实验中心同事的支持和帮助，在此一并表示感谢。

编者

目 录

出版说明

前言

第1章 软件工程基础	1
1.1 软件工程概述	1
1.1.1 软件的特征	1
1.1.2 软件危机	2
1.1.3 软件工程	5
1.1.4 软件生命周期	6
1.1.5 生命周期模型	7
1.1.6 软件开发方法	9
1.2 传统方法学	12
1.2.1 需求分析	13
1.2.2 结构化设计	19
1.2.3 软件构造	21
1.2.4 软件测试和调试	22
1.2.5 软件维护	29
1.3 计算模式简述	29
1.3.1 集中式计算模式	30
1.3.2 C/S 计算模式	30
1.3.3 B/S 计算模式	31
1.4 习题	32
第2章 数据结构及其应用	35
2.1 数据结构基本概念	35
2.2 线性数据结构	37
2.2.1 顺序表	37
2.2.2 线性链表	42
2.2.3 栈	50
2.2.4 队列	55
2.2.5 多维数组	62
2.3 非线性数据结构	66
2.3.1 树和二叉树	67
2.3.2 二叉树的存储及遍历	69
2.3.3 图的基本概念	71
2.3.4 图的存储方式	72

2.3.5	图的遍历方法	75
2.3.6	树和图的应用	77
2.4	查找和排序	81
2.4.1	查找的基本概念	81
2.4.2	静态查找技术	81
2.4.3	动态查找技术	84
2.4.4	排序的基本概念	88
2.4.5	常用排序方法	89
2.5	习题	93
第3章	操作系统及相关程序设计	97
3.1	操作系统概述	97
3.1.1	操作系统的发展和类型	97
3.1.2	常用操作系统简介	99
3.2	操作系统的资源管理	104
3.2.1	进程管理	104
3.2.2	存储管理	107
3.2.3	设备管理	109
3.2.4	文件管理	111
3.2.5	用户接口	114
3.3	Windows 操作系统应用程序设计	115
3.3.1	进程管理应用程序设计	115
3.3.2	存储管理应用程序设计	125
3.3.3	设备管理应用程序设计	129
3.3.4	文件管理应用程序设计	134
3.3.5	用户接口程序设计	138
3.4	习题	148
第4章	数据库及应用程序开发	152
4.1	数据库技术基础	152
4.1.1	数据模型	152
4.1.2	规范化理论	156
4.1.3	关系数据库标准语言 SQL	161
4.2	数据库设计	174
4.2.1	需求分析与概念设计	174
4.2.2	逻辑结构设计	175
4.2.3	数据库物理设计及实施	180
4.3	数据库编程	182
4.3.1	数据库连接技术	182
4.3.2	利用 Visual C++ 开发应用系统	183
4.3.3	编程实例	187

4.4 习题	197
第5章 网络软件开发技术	200
5.1 Internet 基础	200
5.1.1 网络协议和体系结构	200
5.1.2 TCP/IP 地址模式	202
5.1.3 Internet 传输层协议	204
5.2 Socket 编程概述	206
5.2.1 Socket 基本概念	206
5.2.2 面向连接的客户服务器编程	208
5.2.3 无连接的客户服务器编程	215
5.3 迭代和并发服务器的设计	217
5.4 使用 MFC 网络编程	222
5.4.1 CAsyncSocket 类	222
5.4.2 CSocket 类	223
5.4.3 CSocket 编程实例	224
5.5 习题	228
第6章 多媒体编程技术	230
6.1 使用高级函数播放声音	230
6.2 媒体控制接口 MCI	232
6.2.1 命令消息接口	234
6.2.2 命令字符串接口	239
6.2.3 MCIWnd 窗口类	242
6.3 Windows GDI 绘图和显示图像	246
6.3.1 使用 Windows GDI 绘制图形	246
6.3.2 位图的显示	253
6.3.3 制作位图动画	255
6.4 使用 ActiveX 的媒体播放	258
6.4.1 多媒体控制控件	259
6.4.2 媒体播放器	259
6.4.3 Flash 对象控件	261
6.5 习题	262
附录	265
附录 A Visual C++ 6.0 Windows 编程基础	265
A.1 Windows 编程的基本概念	265
A.2 基于文档视图结构的 MFC 应用程序	266
A.3 基于对话框的应用程序	275
附录 B 实验	278
实验1 利用顺序表实现学生信息管理	278
实验2 利用单链表实现学生信息管理	280

实验 3 二叉树的生成和遍历	282
实验 4 学生成绩的查找和排序	284
实验 5 编写 Windows 多线程程序	285
实验 6 编写 Windows 文件读写程序	286
实验 7 数据库的建立和操作	287
实验 8 数据库编程	289
实验 9 消息回声	290
实验 10 访问计数器	291
实验 11 MP3 播放器	294
实验 12 Visual C++ 下绘制图形	294
实验 13 BMP 位图动画	295
参考文献	296

第1章 软件工程基础

计算机程序是根据用户需求用计算机语言描述的适合计算机执行的指令（或语句）序列。而软件是比程序外延更大的一个概念。计算机软件是计算机程序、数据及相关文档的集合。其中，数据是计算机处理的对象；文档是与程序的开发、维护和使用有关的图文资料。

1.1 软件工程概述

当电子邮件、网购、微信、二维码、百度检索已成为人们日常生活和工作的一部分时，这预示着人们已经身处以计算机为核心的网络信息时代。小到文档排版、游戏娱乐，大到年月科考、火星探测，所有这一切都离不开计算机软件。

著名计算机科学家 Roger S. Pressman 在谈到软件时是这样叙述的：“计算机软件已经成为一种驱动力。它是进行商业决策的引擎；它是现代科学研究和工程问题寻求解答的基础；它也是鉴别现代产品和服务的关键因素。它被嵌入在各类系统中：交通、医疗、电信、军事、工业生产过程、娱乐、办公……难以穷举。软件在现代社会中确实是必不可少的。而我们在进入 21 世纪，软件将成为从基础教育到基因工程的所有领域新进展的驱动器。”然而软件的生产与硬件的生产却大不相同。下面先看看软件的特征。

1.1.1 软件的特征

软件在设计、开发、生产、维护和使用等方面都与硬件有着明显的差异。了解这些差异有助于深入了解软件，认识软件开发。

1. 软件是一种逻辑的产品

硬件是看得见、摸得着的物理部件或设备。在研制硬件产品时，人的创造性活动表现在把原材料转变成有形的物理产品。例如，研制出一种新型的计算机主板、CPU 芯片、可重写的光盘、高速路由器等。

而软件产品是以程序和文档的形式存在的，通过在计算机上运行来体现它的作用。在研制软件产品的过程中，人们的生产活动表现在：要创造性地抽象出问题的求解模型，然后根据求解模型写出程序，最后经过调试、运行程序得到求解问题的结果。整个生产、开发过程是在无形的方式下完成的，其“能见度”极差，这给软件开发、生产过程的管理带来了极大的困难。

2. 软件产品质量的体现方式不同

质量的体现方式不同表现在设计和生产两个方面。硬件产品设计定型后可以批量生产，产品质量通过质量检测体系可以得到保障。但是生产、加工过程一旦失误，硬件产品可能就会因为质量问题而报废。而软件产品不能用传统意义上的制造进行生产，就目前软件开发技术而言，软件生产还是“定制”的，只能针对特定问题进行设计或实现。但是软件产品一

一旦实现后，其生产过程只是复制而已，而复制生产出来的软件质量是相同的。设计出来的软件即使出现质量问题，产品也不会报废，通过修改、测试，还可以将“报废”的软件“修复”，投入正常运行。可见，软件的质量保证机制比硬件具有更大的灵活性。

3. 软件产品的成本构成不同

硬件产品的成本构成中有形的物质占了相当大的比重（如工厂、矿山、设备、运输机械、原材料等），人力资源占的比例相对较小。就硬件产品的生存周期而言，成本构成中设计、生产环节占绝大部分，而售后服务只占少部分。

软件生产主要靠脑力劳动。软件产品的成本构成中人力资源占了相当大的比重。软件产品的生产成本主要在开发和研制。研制成功后，产品生产就简单了，通过复制就能批量生产。

随着计算机应用领域的不断拓宽，对软件的需求越来越多，软件的生产费用也在不断增加，导致生产成本不断增加。

4. 软件产品的失败曲线不同

硬件产品存在老化和折旧问题。当一个硬件部件磨损时，可以用一个新部件去替换它。硬件会因为主要部件的磨损而最终被淘汰。

对于软件而言，不存在折旧和磨损问题，如果需要，可以永久使用下去。但是软件故障的排除要比硬件故障的排除复杂得多。软件故障主要是因为软件设计或编码存在错误所致，必须重新设计和编码（设计、测试和调试）才能解决问题。

软件在其开发初始阶段存在很高的失败率，这是由于需求分析不切合实际，或设计错误等引起的。当开发过程中的错误被纠正后，其失败率便下降到一定水平并保持相对稳定，直到该软件被废弃不用。在对软件进行大的改动时，也会导致其失败率急剧上升。

5. 大多数软件仍然是定制生产的

硬件产品一旦设计定型，其生产技术、加工工艺和流程管理也就确定下来，这样便于实现硬件产品的标准化、系列化成批生产。由于硬件产品具有标准的框架和接口，不论哪个厂家的产品，用户买来都可以集成、组装和替换使用。

尽管软件产品复用是软件界孜孜不倦追求的目标，在某些局部范围内几家领军软件企业也建立了一些软件组件复用的技术标准，如 OMG 的 CORBA、Microsoft 的 COM、SUN（后被 Oracle 收购）的 Java EE，但是目前还做不到大范围使用软件复制品。大多数软件仍然是为特定任务或用户定制的。

由于软件的特殊性，在设计、开发、生产、维护和使用等方面也需要考虑不同的策略和方法。

1.1.2 软件危机

从 20 世纪 60 年代开始，软件界在感受计算机应用造福人类的同时，更经常遭受软件危机的袭扰。

以 IBM 公司的 OS/360 操作系统为例。它共有 4000 多个模块、100 万行指令，共投入人力 5000 名，耗资 5 亿美元，但在交付使用的系统中仍找出 2000 个以上的错误。

在客观上，软件不同于硬件。软件开发实质上是逻辑思维的过程，在写出程序并在计算机上运行之前，软件开发的进展情况难于掌握，质量也难于评价，因此管理软件开发过程十

分困难。同时，软件规模和复杂度呈指数级增长。成百上千人共同开发一个大型系统时，大量的通信、后勤工作成为问题。这常常是造成软件开发失败多、费用高的主要原因。人们面临的不光是技术问题，更重要的是管理问题。管理不善必然导致失败。

经研究发现，研制软件系统需要投入大量的人力、财力和物力，但是系统的质量却无法保证。开发软件所需的高成本与软件产品的低质量之间存在尖锐的矛盾，致使软件开发陷入不可自拔的恶性循环之中，这种现象又被称为“软件危机”（Software Crisis）。

1. 软件危机的表现

软件危机主要体现在：

1) 软件开发进度难以预测。拖延工期几个月甚至几年的现象并不罕见，这种情况降低了软件开发组织的信誉。据一项研究统计结果表明：只有 15% 的项目是按计划进度完成的。

2) 软件开发成本难以控制。据同样的研究统计结果表明：仅有 10% 的项目是按费用计划完成的。

3) 用户对软件产品的功能要求难以满足。表 1-1 列出了在美国因开发的软件不能满足用户需求所造成的经济损失的统计资料（1988 年）。

表 1-1 开发软件系统失败所造成的经济损失统计表（1988 年）

系统名称	估计最低费用/百万美元	系统名称	估计最低费用/百万美元
美国矿产部系统	15	先进后勤系统	490
美利坚银行	65	联合教育与软件公司	650
联航订票系统	145	美国空军 B-1BEW 系统	1200
美国海军自动财务系统	446	英国猎迷机载预警系统	2200

4) 软件产品的质量无法保证，系统中的错误难以消除。软件是逻辑产品，质量问题很难以统一的标准度量，因而造成质量控制困难。据统计数据表明，在美国，软件开发项目的开发时间平均超出计划时间的 50%。软件项目越大，情况就越坏。所有大型系统中，大约有 3/4 的系统有运行问题，要么不像预料的那样起作用，要么就根本不能使用。1979 年，美国 US Government Accounting Office 对政府开发的 9 个软件项目进行了调查，调查结果见表 1-2。随着软件开发技术的发展和科学化、规范化的软件项目管理，使得情况有所改善，但是问题仍然严重。例如，英国国防部猎迷机载预警系统的需求和设计修改了无数次，还开发了复杂的多处理系统功能软件，但该项目最终还是被取消。日本的第五代计算机计划，投入 50 亿美元后，因软件原因于 1993 年被迫下马。

表 1-2 软件项目调查结果表

调查结果	项目经费（百万美元）	所占比例/%
付了钱，但系统从没有交付	2.0	28.8
系统交付了，但无法顺利使用	3.2	47.3
系统大部分重做或放弃后才使用	1.3	19.2
经过修改后使用	0.2	3
系统交付后就直接使用	0.1	1.7

5) 软件产品难以维护。软件产品本质上是开发人员的逻辑思维活动的代码化描述，他人难以理解和替代。开发过程中因规范、标准、编程风格、文档资料、检测手段等很难统一，导致开发出的软件系统可维护性差。

6) 软件通常缺少文档资料。软件的文档是开发组织和用户之间权利和义务的合同书，是系统总体设计者向开发人员下达的任务书，是系统维护人员的技术指导手册，是用户的操作说明书。缺乏必要的文档或者文档不合格，将给软件开发、维护带来严重的后果。

7) 软件成本不断提高。由于微电子技术的进步和生产自动化程度的提高，硬件的成本逐年下降，而软件的开发却需要大量的人力，软件规模和数量的增长更使软件成本不断上升。

8) 软件开发生产率的提高速度难以满足社会需求的增长率。

2. 产生软件危机的原因

Standish Group 在 1995 年作了大量的调查研究后得出软件项目失败的原因，并按其重要程度进行了排序，如表 1-3 所示。

表 1-3 软件项目失败的主要原因		
序 号	主要因素	所占比例/%
1	Incomplete Requirements (不完整的需求)	13.1
2	Lack of User Involvement (缺乏用户参与)	12.4
3	Lack of Resources (缺乏资源)	10.6
4	Unrealistic Expectations (不实际的期望)	9.9
5	Lack of Executive Support (缺乏执行的支持)	9.3
6	Changing Requirements & Specifications (需求和规格的变化)	8.7
7	Lack of Planning (缺乏计划)	8.1
8	Do not Need It Any Longer (不再需要)	7.5
9	Lack of IT Management (缺乏 IT 管理)	6.2
10	Technology Illiteracy (技术落后/技术盲区)	4.3
11	Others (其他)	9.9

从表的 1、2、4、6、8 可以看出，用户需求不稳定、不清晰、不完整是项目失败的主要原因。软件开发方法的研究应针对项目失败的原因系统地提出解决办法。

3. 解决软件危机的途径

如何解决软件危机，如何提高软件的生产效率和软件产品的可维护性，一直是困扰软件界的难题。人们经过长期的研究和探索，开始找出解决软件危机的途径。

解决软件危机要从组织管理措施和技术方法两个方面综合考虑，才能从根本上解决问题，这两个条件缺一不可。

硬件生产和软件生产的效率之所以有巨大的差别，除了这两类产品的特征因素外，主要原因之一是组织管理方式。硬件生产早已采用了现代化工程管理方式对生产环节各要素实现统一管理、优化调度、最佳组合；充分发挥有限资源的最大潜能；机械化、自动化的生产线取代了人的体力操作，严密的质量检测仪器取代了人的脑力劳动等。而软件生产还是个体劳动方式，自产自销，手工劳动，不成规模，生产效率低下，质量检测还是凭个人经验。

软件开发不应是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。软件生产也必须采用现代化、社会化的组织管理方式，必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法，特别要吸取几十年来人类从事计算机硬、软件研究和开发的经验教训。

软件开发本身是高度智力密集型劳动，使用先进、得心应手的工具可以“放大”人的智能和体能。在软件开发过程中，要不断引入新技术、新开发工具，将软件工程方法学与自动化软件开发工具相结合，使得开发与维护过程中繁杂重复性的手工劳动能够智能化、自动化，从而提高软件的开发效率和开发质量。

总之，为了解决软件危机，既要有技术措施（方法和工具），又要有严谨的组织管理措施。软件工程正是从管理和技术两方面，研究如何更好地开发和维护软件的一门新兴学科。

1.1.3 软件工程

1. 软件工程的定义

1968年，北大西洋公约组织在联邦德国召开的一次专门研讨解决软件危机的国际会议上，正式提出了“软件工程”的概念。

会上，德国人 Fritz Bauer 提出，“软件工程是建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法”。

1993年，电气和电子工程师学会（Institute Of Electrical And Electronic Engineers, IEEE）给出的定义是“将系统的、规范的和可度量的方法应用于软件的开发、运行和维护，即将工程化应用于软件中”。

GB/T 11457—2006《信息技术 软件工程术语》对软件工程的定义是“应用计算机科学理论和技术以及工程管理的原则和方法，按预算和进度，实现满足用户要求的软件产品的定义、开发、发布和维护的工程或进行研究的学科”。

2. 软件工程的目标

软件工程的目标就是在给定成本和进度的前提下，开发出具有有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可操作性且满足用户需求的产品。其基本目标是：

1) 提高软件的生产效率，更快、更多地开发软件，按时完成开发任务。

2) 达到软件要求的功能。

3) 取得较好的软件性能，包括易于移植、维护，可靠性高等。

4) 降低软件的开发成本。

3. 软件工程的研究内容

基于软件工程的目标，软件工程研究的主要内容包括软件开发技术和软件工程管理两方面的内容。

(1) 软件开发技术

软件开发技术包括软件开发方法学、开发过程、开发工具和软件工程环境，其主体内容是软件开发方法学。软件开发方法学是根据不同的软件类型，按不同的观点和原则，对软件开发中应遵循的策略、原则、步骤和必须产生的文档资料作出规定，从而使软件的开发能够规范化和工程化，以克服早期手工方式生产时的随意性和非规范性。

软件开发过程是把用户要求转化为软件产品的过程。此过程包括把用户要求转化为软件需求,把软件需求转化为设计,用代码来实现设计,对代码进行测试,有时还包括安装和验收等。

软件工具是指一些计算机程序,用来帮助开发、测试、分析或维护计算机程序或它的文件,如编辑程序、流程图绘制程序、反编译程序等。

软件开发环境 (Software Eevelopment Environment, SDE) 指支持软件产品开发的软件系统。它由软件工具和环境集成机制构成,前者用以支持软件开发的相关过程、活动和任务,后者为工具集成和软件的开发、维护及管理提供统一的支持。

(2) 软件工程管理

软件工程管理要求按照预先制订的计划、进度和预算执行,以实现预期的经济效益和社会效益。统计表明,多数软件项目的失败并不是由于技术原因造成的,而是由管理不当造成的。

软件工程管理包括软件管理学、软件工程经济学、软件心理学等内容。

软件管理学包括人员组织、进度安排、质量保证、配置管理(对软件开发中产生的各种实体的功能、特性等进行标识、说明,并控制它们的变更)和项目计划等。

软件工程经济学是研究软件开发中成本的估算、成本效益分析的方法和技术,用经济学的观点来研究、分析如何有效地开发、发布软件产品和吸引用户使用这一产品。

软件心理学从个体心理、人类行为、组织行为和企业文化等角度来研究软件管理和软件工程。

1.1.4 软件生命周期

通常将软件产品从提出开发要求开始,经过需求分析、设计、编码、测试、使用,直到该软件产品被淘汰为止的整个过程称为软件生命周期,也称作软件生存周期。

软件生命周期分为软件定义、软件开发和运行维护(也叫软件维护)3个时期,每个时期又进一步划分成若干阶段。

1. 软件定义时期

软件定义时期的任务是:确定软件开发项目必须完成的总目标;确定项目的可行性;导出实现项目目标应采取的策略及系统必须完成的功能;估计完成该项目需要的资源和成本并制订项目进度表。这个时期的工作通常又被称为系统分析,由系统分析员负责完成。软件定义时期通常有3个阶段:问题定义、可行性研究和需求分析。

问题定义回答“要解决的问题是什么”,通过对客户的访问调查,应完成关于问题的性质、工程目标和工程规模的书面报告。

可行性研究回答“对于上一个阶段所确定的问题是否有行得通的解决方法”,研究软件的范围,探究是否值得去做,是否能够做到。

需求分析阶段准确地确定“为了解决这个问题,目标系统必须做什么”,主要是确定目标系统必须具备哪些功能。需求分析阶段确定的系统逻辑模型是以后设计和实现目标系统的基础,因此必须准确完整地体现用户的需求。这一阶段的一项重要任务是用文档准确地记录对目标系统的需求,这一文档称为需求规格说明书或系统分析说明书。

2. 软件开发时期

软件开发时期的主要任务是设计和实现前一时期定义的软件。它通常由总体设计、详细

设计、编码和测试4个阶段组成，其中前两个阶段又称为系统设计，后两个阶段称为系统实现。

总体设计，“概括地说应该怎样实现目标系统”，又称概要设计。总体设计的一项重要任务是设计软件的体系结构，也就是确定软件由哪些部分组成以及这些部分之间的关系。每一个部分称为一个模块。概要设计应给出几种方案以及每种方案的优缺点，从中选出最佳方案。

总体设计的解决方案比较概括和抽象，而详细设计的任务就是把解决方案具体化，把概括的模块再细分成更具体的模块，给出每个模块的数据结构和算法。详细设计也称模块设计。这一阶段完成的主要文档是详细规格说明书，它类似于工程领域中的工程蓝图。注意详细设计的任务仍不是编写程序。

编码阶段写出正确的、容易理解的和容易维护的程序模块。

测试阶段是为检验程序中的错误而去运行程序。为了发现系统中的错误，应竭力设计能暴露错误的测试用例。测试用例是由测试数据和预期结果构成的。一个好的测试用例是极有可能发现至今为止尚未发现的错误的测试用例。如果发现了错误，就应该找到错误的原因并更正错误，称为调试。发现的错误越多，程序中的错误越少，软件的可靠性就越高。所以测试的数据和过程应经过设计，以让错误暴露的更多。

3. 软件维护时期

维护时期的主要任务是使软件持久地满足用户的需要。当软件在使用过程中发现错误时应该加以改正；当环境改变时应该修改软件以适应新的环境；当用户有新的要求时应该及时改进软件以满足用户的新需要。维护时期一般不再划分阶段，但每一次维护活动都相当于一次简化定义和开发过程。

生命周期阶段的划分和实施的顺序、要求并不唯一。当然，实施过程的不同，对软件开发过程会产生影响。把生命周期划分成哪些阶段以及各个阶段的执行顺序的规划称为生命周期模型。

1.1.5 生命周期模型

生命周期模型也叫软件工程模型、软件开发模型或过程模型，它是一个框架，从需求定义到使用终止，跨越整个生存期的系统开发、操作和维护。典型的生命周期模型有瀑布模型、快速原型模型、增量模型、螺旋模型等。为描述方便，常常将问题的定义和可行性研究概括到用户需求中，把总体设计和详细设计合并为设计。

1. 瀑布模型

瀑布模型是1970年由著名软件工程专家Winston Royce提出的，直到20世纪80年代早期，它一直是唯一被广泛采用的软件开发模型。

瀑布模型将软件生存周期中各个阶段依线性顺序连接。每个阶段的结果是一个或多个经过核准的文档。直到上一个阶段完成，下一个阶段才能启动（见图1-1）。瀑布模型并不是简单的线性模型，它允许开发活动的多个反复。

瀑布模型的优点是强迫开发人员采用规范化的方法；强调文档的作用，使软件维护比较容易；每个阶段都要严格地验证，减少随意性。它的优点也是它的缺点，一是按照规范和经过验证的文档开发，使得它难以响应用户需求的变更；二是如果软件需求不太明确，则无法进行软件开发工作。

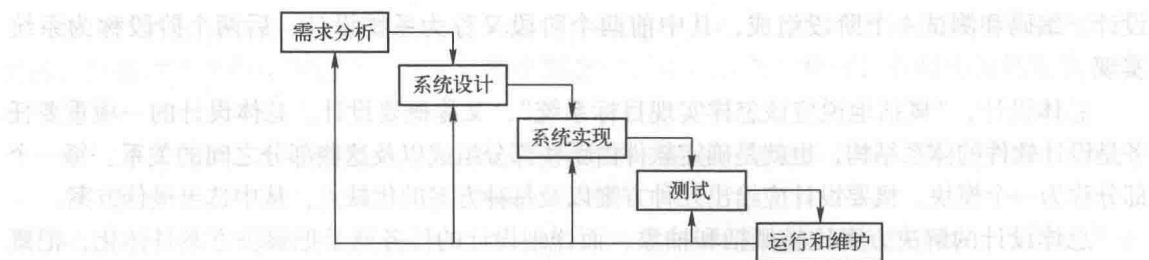


图 1-1 瀑布模型

2. 快速原型模型

快速原型是快速建立起来的可以在计算机上运行的程序，它所完成的功能往往是最最终产品功能的一个子集。快速原型模型首先快速建立系统的原型，让用户试用，了解目标系统的概貌。用户试用后提出修改意见，开发人员按照用户意见快速地修改原型系统，直到用户确认原型系统能完成所需的功能。开发人员据此编写规格说明书，根据这份文档完成满足用户需求

快速原型模型的关键在于尽可能快速地构造出软件原型，一旦确定了客户的真正需求，所建造的原型将被丢弃。因此，原型系统的内部结构并不重要，重要的是必须迅速建立原型，随之迅速修改原型，以反映客户的需求。它具有快速了解用户需求、缩短开发周期和降低开发成本等优点。

快速原型模型可以降低因产品不能满足用户需求所带来的风险，但不能应对所有风险。

3. 增量模型

增量模型也称渐增模型。在增量模型中，软件被作为一系列的增量构件来设计、实现、集成和测试。在使用增量模型时，第一个增量往往是实现基本需求的核心构件。该核心构件交付用户使用后，经过评价形成下一个增量的开发计划，它包括对核心构件的修改和增加具有新功能的构件。这个过程在每个增量发布后不断重复，直到产生最终的完善产品。

增量模型的优点是能够在短时间内向用户提交可完成部分工作的产品，减少全新软件给用户带来的不适应。其缺点是每一个新的增量构件必须不能破坏原来已经开发出的产品，这就要求在实现各个构件前就要全部完成需求分析和系统的设计，对需求和设计的要求高。

4. 螺旋模型

软件开发带有风险，如不满足用户需求，技术储备不足，不能按时完工，成本超出预算，市场竞争，管理不善等，这些会使项目失败或造成经济损失。

1988 年，Barry Boehm 发表了“螺旋模型”，它不是将软件过程用一系列活动和活动间的回溯来表示，而是将过程用螺旋线表示，如图 1-2 所示。在螺旋线中，每个周期表示软件过程的一个阶段。最里面的周期与系统的定义和可行性有关，下一个周期与需求定义有关，再下一个与系统设计有关等。每个周期有制订计划、风险分析、开发实施和用户评估 4 个环节。

螺旋模型的每个周期首先是确定该阶段的目标以及为完成这些目标选择的方案及其约束条件；然后从风险角度分析方案的开发策略，努力排除各种潜在的风险，有时需要通过建造原型来完成。如果某些风险不能排除，该方案立即终止，否则启动下一个开发步骤；随后的开发实施阶段相当于局部的瀑布模型；最后，评价该周期的结果，并设计下一个周期的计