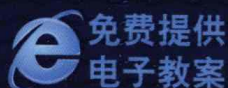




普通高等教育“十一五”国家级规划教材



免费提供
电子教案

高等院校规划教材
计算机基础教育系列

软件开发技术基础

第2版

赵英良 主编
仇国巍 薛涛 卫颜俊 等编著
冯博琴 审



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十一五”国家级规划教材
高等院校规划教材·计算机基础教育系列

软件开发技术基础

第2版

	赵英良	主编	
仇国巍	薛涛	卫颜俊	
李尊朝	刘志强	崔舒宁	编著
	冯博琴	审	

机械工业出版社

本书是根据教育部非计算机专业计算机基础课程教学指导分委员会《关于进一步加强高校计算机基础教学的意见》中软件类课程的要求,结合非计算机专业软件开发的特点组织编写而成的。

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、数据结构及应用、操作系统及相关程序设计、数据库及应用
程序开发、网络编程技术、多媒体编程技术等。书中附录 A 为 Visual
C++ 6.0 Windows 编程基础,附录 B 为实验指导。本书以 C++ 为编程语
言,内容实用,条理清晰,每章均配有例题和习题,旨在锻炼学生的软件
开发能力。

与第 1 版相比,本书内容系统性更强,例题介绍更加简洁、实用,易
于实现。本书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。

图书在版编目 (CIP) 数据

软件开发技术基础/赵英良主编. —2 版. —北京:机
械工业出版社, 2009. 4

普通高等教育“十一五”国家级规划教材. 高等院校
规划教材. 计算机基础教育系列

ISBN 978 - 7 - 111 - 26532 - 0

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

中国版本图书馆 CIP 数据核字 (2009) 第 033869 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: 张宝珠 责任编辑: 张宝珠 谷玉春

版式设计: 张世琴 责任校对: 魏俊云

封面设计: 李 瞳 责任印制: 乔 宇

北京京丰印刷厂印刷

2009 年 4 月第 2 版 · 第 1 次印刷

184mm × 260mm · 20.25 印张 · 499 千字

0 001—3 000 册

标准书号: ISBN 978 - 7 - 111 - 26532 - 0

定价: 34.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379739

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。另外，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师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！

机械工业出版社

前 言

进入新世纪,计算机硬件技术和软件技术都取得了突飞猛进的发展,促使我们必须提高计算机教育水平,加强大学生的计算机基础知识和应用能力。基于此,2005年,我们对“软件技术基础”的内容进行了大幅度的改革,从以原理介绍为主改为以软件开发训练为主,旨在加强对大学生软件开发能力的培养,编写出版了“软件技术基础”课程的新教材——《软件开发技术基础》。本书第1版得到不少高校和专家的认可,2006年,本书被列为普通高等教育“十一五”国家级规划教材。2007年,“软件开发技术基础”课程被评为国家级精品课程。

在第1版两年多的使用中,我们发现了其中的一些不足,同时也采纳了一些专家和读者的建议。在第1版的基础上删掉了原来过于笼统、庞杂的内容,对原来零散的内容进行了梳理、补充,使第2版的内容更加系统和实用。具体的修改如下:

第1章删掉了原“软件工程技术”一节的大部分内容,重点介绍了软件开发各阶段的任务、方法和工具。

第2章补充了稀疏矩阵的处理和字符串处理的内容和实例。

第3章补充了操作系统原理的介绍,加强了内容的系统性,修改和补充了部分例题,使同学们更容易练习和实践。

第4章修改了部分例题,使编程实现更加简洁。

第5章删掉了“从Web服务器下载文件”和“多线程文本聊天系统”两节的内容。修改了例题的解法,使条理更清楚。增加了通过控件实现网络应用的一般方法。通过本章的学习,使读者快速了解网络通信的基本原理,并能建立简单的应用系统。

第6章删除了原6.1节(多媒体基础)的大部分内容,第2版中的6.1节(多媒体编程基础)重点介绍与多媒体应用程序有关的内容,如MCI接口、ActiveX等。后面比较系统地介绍了如何通过MCI接口对声音进行控制、Visual C++ 6.0环境下图形的绘制、位图的显示以及在Visual C++中如何使用多媒体控件等。

第2版还对实验进行了修改:增加了指导性的内容,删除了教学中不经常使用的实验。附录A中的“Visual C++ 6.0 Windows编程基础”仅保留了最常用的内容。

本书第2版由赵英良任主编,并改编了第6章,刘志强、仇国巍、卫颜俊、李尊朝、薛涛老师分别改编了第1~5章。国家级教学名师冯博琴教授在百忙中审阅了书稿。本书得到了西安交通大学计算机教学实验中心许多同事的支持和帮助,在此一并表示感谢。

本书还配有电子教案,供广大读者学习和交流使用,下载网址为 www.cmpedu.com。

作 者

目 录

出版说明

前言

第 1 章 软件与软件工程	1
1.1 软件概述	1
1.1.1 软件的特征	1
1.1.2 软件的发展	2
1.1.3 软件危机	5
1.2 软件工程	8
1.2.1 软件生存周期	8
1.2.2 软件工程模型	9
1.2.3 软件的开发过程	11
1.2.4 软件开发方法	20
1.3 计算模式	23
1.3.1 集中式计算模式	24
1.3.2 客户端/服务器计算模式	24
1.3.3 基于 Web 工程设计基础	26
1.4 习题	29
第 2 章 数据结构及其应用	31
2.1 数据结构基本概念	31
2.2 线性数据结构	33
2.2.1 顺序表	33
2.2.2 线性链表	39
2.2.3 堆栈	48
2.2.4 队列	53
2.3 非线性数据结构	60
2.3.1 多维数组	60
2.3.2 二叉树的基本概念	64
2.3.3 二叉树的存储及遍历	66
2.3.4 图的基本概念	68
2.3.5 图的存储方式	69
2.3.6 图的遍历方法	72
2.3.7 树和图的应用	75
2.4 查找和排序	82
2.4.1 查找的基本概念	82
2.4.2 静态查找技术	83
2.4.3 动态查找技术	85

2.4.4 排序的基本概念	90
2.4.5 常用排序方法	91
2.5 习题	95
第3章 操作系统及相关程序设计	99
3.1 操作系统概述	99
3.1.1 操作系统的发展	99
3.1.2 操作系统的功能	101
3.1.3 操作系统的分类	102
3.1.4 常用操作系统简介	104
3.2 进程管理	107
3.2.1 CPU 与进程管理	107
3.2.2 进程程序设计	110
3.2.3 线程程序设计	115
3.2.4 定时器的应用	118
3.3 存储管理应用程序设计	120
3.3.1 存储管理	120
3.3.2 Windows XP 的存储管理	122
3.3.3 虚拟存储管理程序设计	123
3.3.4 动态链接库的建立和使用	125
3.4 设备管理应用程序设计	127
3.4.1 设备管理	127
3.4.2 设备管理程序设计	129
3.5 文件管理应用程序设计	134
3.5.1 文件管理	135
3.5.2 文件管理程序设计	137
3.6 人机接口	143
3.6.1 人机接口的概念	143
3.6.2 图形界面程序设计	143
3.7 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	160
4.2 数据库设计	171
4.2.1 需求分析与概念设计	172
4.2.2 逻辑结构设计	172
4.2.3 数据库物理设计及实施	176
4.3 数据库应用程序编程	178
4.3.1 数据库连接技术	178
4.3.2 利用 Visual C++ 开发应用系统	179

4.3.3 编程实例	183
4.4 习题	193
第5章 网络编程技术	196
5.1 Internet 基础	196
5.1.1 网络协议和体系结构	196
5.1.2 TCP/IP 地址模式	198
5.1.3 Internet 传输层协议	200
5.2 Socket 概述	201
5.2.1 Socket 简介	201
5.2.2 初始化 WinSock	202
5.2.3 协议地址及转换函数	203
5.2.4 创建和关闭 Socket	205
5.3 面向连接的客户端/服务器编程	205
5.3.1 服务器端 API	205
5.3.2 客户端 API	208
5.3.3 数据传输 API	208
5.4 无连接的客户端/服务器编程	213
5.5 迭代和并发服务器的设计	220
5.6 MFC 网络编程	225
5.7 习题	241
第6章 多媒体编程技术	243
6.1 多媒体编程基础	243
6.1.1 媒体控制接口	243
6.1.2 ActiveX 技术简介	244
6.1.3 DirectX 技术简介	245
6.2 音频的播放与处理	247
6.2.1 命令消息接口	247
6.2.2 命令字符串接口	252
6.2.3 WAV 文件结构	253
6.2.4 音效处理	255
6.3 图像处理	259
6.3.1 使用 Windows GDI 绘制图形	259
6.3.2 位图的显示	265
6.3.3 BMP 位图文件的操作	268
6.4 动态图像	275
6.4.1 使用 ActiveX 控件播放视频文件	275
6.4.2 制作位图动画	276
6.5 习题	279
附录	282
附录 A Visual C++ 6.0 Windows 编程基础	282
A.1 Windows 编程的基本概念	282

A.2 基于文档视图结构的 MFC 应用程序	283
附录 B 实验指导	293
实验 1 利用顺序表实现学生信息管理	293
实验 2 利用单链表实现学生信息管理	296
实验 3 二叉树的生成和遍历	298
实验 4 学生成绩的查找和排序	300
实验 5 编写可调用 Windows 系统命令的程序	301
实验 6 编写模仿“我的电脑”功能的程序	302
实验 7 数据库的建立和操作	303
实验 8 数据库编程	305
实验 9 消息回声	305
实验 10 访问计数器	307
实验 11 媒体播放器	309
实验 12 BMP 位图动画	313
参考文献	314

第 1 章 软件与软件工程

即使是对于非计算机专业的学生而言，不会开发软件也是不可想象的。但是要开发一款软件，特别是涉及复杂应用背景的大型软件，如基于 Web 的酒店管理系统（包括前台的预定、登记、自动记账、餐饮管理、客房管理、采购计划、决策支持等功能），事情就不是那么简单了。

本章讲述软件的特征、开发软件的方法、软件开发技术和用工程化的方式生产软件等问题。

1.1 软件概述

随着以计算机为核心的网络信息时代的到来，小到个人电子文档排版、下棋娱乐，大到指挥控制火星探测器登陆火星，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。”

1.1.1 软件的特征

软件及其蕴涵的精髓曾经是计算机专业人员拥有的一种文化素养。随着软件地位的提高，如今已经是对整个社会产生重要影响的文化素养了。因此，有必要充分认识软件的特征。

1. 软件是一种逻辑的产品，与硬件产品有本质的区别

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，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、开发过程是在无形化方式下完成的，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与硬件产品不同

质量体现方式不同表现在两个方面。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与硬件产品不同

硬件产品的成本构成中有形的物质占了相当大的比重。例如，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，对软件的需求也越来越多，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与硬件产品不同

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，如果需要的话可以永远使用下去。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的错误所致，必须重新设计和编码（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，这主要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。在软件进行大的改动时，也会导致失败率急剧上升。

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，例如，OMG 的 CORBA，Microsoft 的 COM，Sun 的 J2EE 等，但是目前还做不到大范围使用软件替代品。大多数软件仍然是为特定任务或用户定制的。

1.1.2 软件的发展

软件的发展与任何其他事物的发展一样，经历了从小到大、从简单到复杂的过程。回顾软件的发展经历，可以看出由应用驱动为内在动力推进软件发展进程是一条软件发展的主线。从软件发展早期的按个人意愿编写代码，到按软件工程的理论生产大型应用系统；从手工作坊式的封闭制作到按标准制式、工业化生产；从注重个人精雕细琢地编写代码的技巧到注重工程项目的管理和开发团队的协作；从为特定应用定制设计到以复用为目标的组件式开发；从由少数软件天才的编程艺术到由广大用户直接参与的应用开发。所有这些发展和进步，充分体现了人类在驾驭数字和计算以造福人类所作出的丰功伟绩。

从 1946 年世界上诞生第一台电子计算机的 60 多年以来, 软件及软件开发技术在理论、观念、目标及技术方法等方面都发生了很大变化。我们以 10 年为间隔, 将其划分为 5 个阶段。分析与讨论每个阶段软件开发技术的发展状况、技术突破、理论成就等。阶段划分与每个阶段软件理论、技术、方法等的发展简况见表 1-1。

表 1-1 软件开发技术发展史

阶段	第一阶段	第二阶段	第三阶段	第四阶段	第五阶段
年代	20 世纪 40 及 50 年代	20 世纪 60 年代	20 世纪 70 年代	20 世纪 80 年代	20 世纪 90 年代
处理对象	二进制代码“0”和“1”	面向应用的高级语言编写的程序	小规模系统的程序设计	大型系统的程序设计	基于 Internet/Web 程序设计
用途	科学计算	科学计算、商务应用、数据处理	商务应用、网络互联技术、数据库应用	关系数据库应用 各领域的计算机应用	网络互联及应用
目的	确定计算机硬件动作的序列	确定计算机指令的动作序列	研究数据结构及算法以寻求处理的时-空平衡	大众化普及应用计算机	通过资源共享和协同处理来提高群组的整体效率
开发方法	经验主义	功能性程序设计	数据结构与算法结构化开发方法	构造系统的方法 用标记工具描述系统 文档是管理系统结构的主要手段	基于部件开发方法 基于 Internet/web 技术的软件开发 部件互连及集成
开发工具	机器语言 汇编语言	汇编语言 高级语言	高级语言 数据库	面向过程语言 基于 PC 的开发工具 早期 CASE 工具	Java 语言 组件技术 基于 Web 开发工具 现代 CASE 工具
认识	计算机的用途就是快速计算	软件应独立于硬件 必须规范程序内部控制流	平衡时-空复杂度 程序执行的停机 强调复杂程序的可读性	软件生命周期 强调开发组协作的重要性	认识系统非功能性需求的重要性 强调在各自条件下追求软件过程改进
发展概况	机器语言编码 汇编语言编码 Fortran 语言编码	高级程序设计语言 编译系统技术 非数值商业应用 出现软件危机	数据独立(数据文件及数据库) 对复杂系统的规格说明	软件工程研究机构 CASE 工具环境 计算机图形界面(GUI)系统	Internet/Intranet 基于 Web 的应用 虚拟现实技术 新技术涌现, 但理论欠缺
重大技术突破	认识时-空关系 迭代-反复算法 思想观念转变	高级语言程序促进计算机应用 功能性程序设计技术	数据结构及算法 各种高级语言 区分了系统软件与应用软件 形成完整的软件系统	RDB 的 C/S 计算模式 GUI 图形用户界面 大型商用 RDBMS	设计模式 浏览器技术 CORBA 技术 DCOM/OLE 技术

(续)

阶段	第一阶段	第二阶段	第三阶段	第四阶段	第五阶段
理论成就	冯·诺依曼理论 图灵测试	用软件工程解决 软件危机 结构化程序设计 理论	数据结构与算法 理论 形式方法 软件工程方法	关系数据库的关 系理论 软件工作量度量 (COCOMO) 软件过程改进模 型 (CMM)	“交互模型”作为 网络计算的理论基 础
目标	用最少资源获得 最大运算能力	扩大程序规模以 适应更复杂的应用	解决软件危机	发展软件复用技 术提高软件生产率	正确的时刻,把 正确的信息安全地 送给需要它的人

1. 第一阶段 (20 世纪 40 及 50 年代)

早期计算机是由只有硬件的“裸机”组成,计算机中不安装任何软件。由于采用电子管作为逻辑电路的元器件,系统稳定性很差。那时人们只关心硬件的性能和技术指标,系统设计与实现是以硬件为中心,软件开发处于从属地位。这个阶段的特征主要表现为:没有系统化的开发方法,程序设计采用封闭式个体开发方式,开发工具是机器语言,追求目标是用最少的资源来获得最大的运算能力。

2. 第二阶段 (20 世纪 60 年代)

由于硬件技术的发展,在电路设计中广泛采用晶体管和集成电路,使计算机的体积大大缩小、系统稳定性提高、价格降低。开始出现以屏幕显示器输出和键盘输入的商用计算机,促进了计算机的产、销量急剧上升。在软件领域,应用领域拓宽,社会对应用软件的需求急剧增加,软件的地位和作用不断提高,从而有力地推动了软件技术和开发方法的改进和发展。这个阶段的特征主要是:开发工具进化为高级程序设计语言,产生了结构化编程的思想和方法,提出了用“软件工程”方法解决“软件危机”的问题。

3. 第三阶段 (20 世纪 70 年代)

这个时期计算机硬件的发展具有里程碑的意义,采用诸如大规模集成电路,分布式技术、网络技术、嵌入式等高新技术,从而极大地提高了计算机系统的处理能力,特别是微处理器的诞生,开创了大众化使用计算机的新时代。该阶段软件的特征是:高性能的计算机系统对软件的功能和性能提出了更高的要求。这一阶段主要表现为:提出了“程序设计 = 数据结构 + 算法”的著名公式,应用开始涉入人工智能的领域,计算机网络互联的技术(电子邮件)、操作系统综合处理功能以及数据库技术等得到大力发展,结构化程序设计方法发展成结构化开发方法(包括结构化分析和结构化设计)。本阶段软件技术发展的目标是解决软件危机。

4. 第四阶段 (20 世纪 80 年代)

这个时期计算机硬件的发展体现在个人计算机和计算机局域网络方面。该阶段软件发展的特征是:个人计算机应用软件异军突起,形成信息产业中新的增长点。伴随微处理器和嵌入式系统的发展,计算机应用开始深入人们的日常工作和生活。旺盛的市场需求和政府企业甚至广大百姓的资金投入,孕育了大量软件企业。市场竞争和市场经济规则促进了软件产品功能更强,质量更高,性能价格比更好。大型机(包括中、小型机)软件向个人计算机上

移植，从而为软件提供了更广阔的发展空间和巨大的用户市场。这一阶段主要表现：由于数据库技术的成熟和个人计算机的普及，计算机的主要用途转向各个领域的数据库系统应用；开发工具的改进具有时代特点：一是供个人计算机系统使用；二是出现计算机辅助软件工程（CASE）工具。为更好地发展软件技术，各工业发达国家相继成立软件工程研究开发基地（如美国国防部支持的在卡内奇·梅隆大学建立软件工程研究所），推出各项重大的软件工程项目；开发技术在客户端/服务器（C/S）计算模式、图形用户界面（GUI）以及大型商用数据库管理系统等方面取得突破；提出关系数据库、软件工作量估计的构造性成本模型（COCOMO）和评价开发组织生产可靠软件能力的软件过程能力成熟度模型（CMM）等软件技术的理论和方法。

5. 第五阶段（20 世纪 90 年代）

当计算技术发展到网络计算，标志着计算机网络时代的到来。

伴随处理分散的海量数据的应用需求，诞生并促进了分布式技术、并行计算、人工神经网络等技术的发展。反映该阶段的软件特征是：数据由原来的字符信息变为多媒体（文字、声音、图像、动画等）信息，多媒体信息的实时处理技术成为研究的热点；计算机系统内部传统的处理模式（单指令流单数据流和单指令流多数据流）受到极大挑战，具有并行处理功能的新型处理模式（多指令流多数据流）倍受关注和厚爱；面向对象的开发方法代表着在新处理环境中提高软件开发效率和质量的方向；信息唾手可得的网络服务软件满足着人们日益增长的应用需求。

主要表现在：互联网得到广泛应用，在网络背景下计算机应用的目的不仅是要提高个人的生产率，而且通过支持跨地区、跨部门、跨时间的群组共享信息协同工作来提高群组、集团的整体生产率；基于组件的开发方法、与系统无关的跨平台性和基于 Internet/Web 技术等成为软件开发的主要特点；开发工具有 Java 语言、组件开发技术、基于 Web 的开发工具以及现代 CASE 工具等；对软件开发的认识已从目标管理转向过程管理；新软件技术不断涌现，发展无法预料，但缺乏相应的理论；在软件技术方面，设计模式、对体系结构的分类与表示方法的研究、浏览器技术、分布式计算环境解决跨平台互操作性等取得巨大成功；理论研究从以研究封闭的系统转向开放的、不断演化的系统；业内有人分析：作为计算理论的基础——图灵机模型已不适合于作为网络计算的基本模型，而提出了“交互模型”；所有技术追求的目标是：“正确的时刻，把正确的信息、安全地送给需要它的人”（Just in time information）。

1.1.3 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、费用高的重要原因。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、物力和财力，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，投资一再追加，令人难以置信。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 1-2 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 1-2 开发软件系统失败所造成经济损失的统计表

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 1-3。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的不少功能软件，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 1-3 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	1.77

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) 软件开发生产效率的提高速度难以满足社会需求的增长率。软件产品“供不应求”的现象致使不能充分利用现代计算机硬件提供的巨大潜力。

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，见表 1-4。

表 1-4 软件项目失败的主要原因

序号	主要因素	所占百分比 (%)
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 项可以看出，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这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如此巨大的差别，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软、硬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，要坚持不断引入新技术、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。

1.2 软件工程

由于软件危机，使人们认识到软件开发有其独特的内在规律，遵循这个规律，才能掌握软件生产的自由。什么是软件生产的规律呢？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。为研究解决软件开发过程中的开发技术、思想、方法、工具等问题，软件工程逐步发展成为一门独立的学科——软件工程学。

1.2.1 软件生存周期

软件生存周期是指一个软件从提出开发要求开始，经过需求分析、设计、制造、调试、使用维护，直到该软件产品被淘汰为止的整个过程。

应该说人们对于软件生存周期的认识是一个不断深化的过程。1995 年国际标准化组织将软件生存周期的活动和任务划分为主要过程、支持过程和组织过程 3 个过程，表 1-5 列出了软件生存期周期的主要活动和任务描述。

表 1-5 软件生存周期的主要活动和任务描述

	过 程 名	主 体	主要活动和任务描述
主要过程	获取	需方	定义、分析需求或委托供方进行需求分析而后认可，招标准备，合同准备以及验收
	供应	供方	评审需求，准备投标，签定合同，制订并实施项目计划，开展评审及评议，交付产品
	开发	开发者	系统需求、结构分析，软件需求、结构分析，软件详细设计，软件编码，软件、系统集成，软件测试、软件合格测试、系统合格测试，软件安装及软件验收支持
	运行	运行者	制订并实施运行计划，运行测试，系统运行，对用户提供帮助和咨询
	维护	维护者	问题和变更分析，实施变更，维护评审及验收，软件移植及软件退役
支持过程	文档编制	开发者	设计文档编制标准，确认文档输入数据的来源及适宜性，文档的评审及编辑，文档发布前的批准，文档的生产与提交、存储和控制，文档维护
	配置管理	主程序员	配置标识与控制，记录配置状态，评价配置，发行管理与交付
	质量保证	开发组织	软件产品的质量保证，软件过程的质量保证以及按 ISO 9001 标准实施的质量体系保证
	验证	项目负责人	合同、过程、需求、设计、编码集成和文档等的验证
	确认	项目负责人	为分析测试结果实施特定的测试，确认软件产品的用途，测试软件产品的实用性
	联合评审	开发组织	实施项目管理评审（项目计划、进度、标准、指南等的评价），技术评审（评价软件产品的完整性等）
	审核	开发组织	检验项目是否符合需求、计划、合同以及规格说明和标准
	问题解决	主程序员	分析和解决开发、运行、维护或其他过程中出现的问题，提出响应对策，使问题得到解决
组织过程	管理	管理者	制订计划、监控和评价计划的实施，涉及有关过程的产品管理、项目管理和任务管理
	基础设施	开发组织	为其他过程所需的硬件、软件、工具、技术、标准以及开发、运行或维护所用的各种基础设施的建立和维护服务
	改进	开发组织	对整个软件生存过程进行评估、度量、控制和改进
	培训	开发组织	制订培训计划，编写培训教材，培训计划的实施