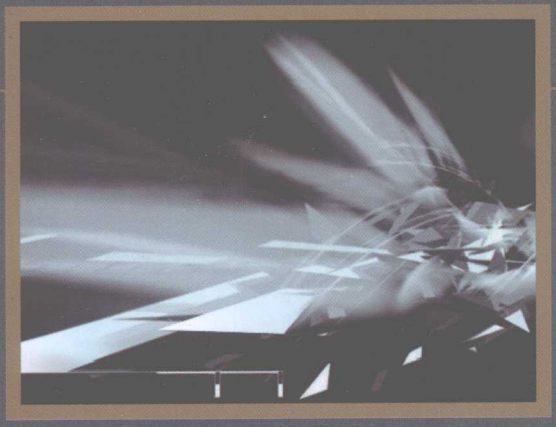
高等学校计算机科学与技术教材

现代软件工程

□ 张泊平　主编



- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精炼，实例丰富
- 可操作性强，实用性突出

清华大学出版社　●　北京交通大学出版社

高等学校计算机科学与技术教材

现代软件工程

张泊平　主编

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书以面向对象软件工程技术为主，重点讲解了软件工程的基本理论、软件工程方法学、面向对象软件工程等方面的内容，并以实际案例分析贯穿始终，对于提高学生的软件开发素养具有一定的指导意义。针对初学者的特点力求理论表述通俗易懂，内容新颖实用，尽量用实例来诠释概念和方法，使读者能够轻松地掌握面向对象软件工程的方法和技能，进而在软件企业很快地进入各种角色。

本书可以作为高等院校计算机本科相关专业高年级学生的教学用书，也可作为专科学生、报考计算机专业的考生、参加国家高等教育自学考试的考生、参加计算机等级考试的考生和计算机专业高级人员的参考用书。

图书在版编目（CIP）数据

现代软件工程／张泊平主编. —北京：清华大学出版社；北京交通大学出版社，2009.8
（高等学校计算机科学与技术教材）
ISBN 978－7－81123－695－8

Ⅰ.现…　Ⅱ.张…　Ⅲ.软件工程－高等学校－教材　Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字（2009）第 119529 号

责任编辑：解　坤
出版发行：清华大学出版社　邮编：100084　电话：010－62776969　http://www.tup.com.cn
　　北京交通大学出版社　邮编：100044　电话：010－51686414　http://press.bjtu.edu.cn
印 刷 者：北京东光印刷厂
经　　销：全国新华书店
开　　本：185×260　印张：21.75　字数：543 千字
版　　次：2009 年 8 月第 1 版　2009 年 8 月第 1 次印刷
书　　号：ISBN 978－7－81123－695－8/TP·505
印　　数：1～4 000 册　定价：33.00 元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。
投诉电话：010－51686043，51686008；传真：010－62225406；E-mail：press@bjtu.edu.cn。

前　言

一、关于本书

软件工程是应用计算机科学、数学及管理科学等原理开发软件的工程。其教育培养目标是让受教育者了解和掌握软件开发中的方法学和工程学知识，并应用于实践。这一目标在现阶段体现了国际化、多元化、本地化和工程化的特点。软件行业深切感到这四方面人才培养的迫切性。自20世纪60年代提出软件工程概念以来，面向对象软件工程技术逐渐成熟，现在已成为计算机科学与技术中的一门重要学科。

本书是作者根据近十年来对软件工程学、面向对象方法的教学研究和实践经验，结合软件开发新技术精心编写的。

二、本书内容

本书由14章组成，主要包括软件工程概述、可行性分析、软件需求分析、软件外部设计——交互设计、软件内部设计、面向对象软件工程基础、统一建模语言、软件编码与实现、软件测试、软件维护、软件复用和构件技术、软件项目计划与管理、软件工程环境、软件工程课程设计等。

三、本书特点

强调实例分析和应用训练是本书的主要特色。

① 内容安排新颖实用，突出“通俗易懂”。本书针对初学者的特点，力求理论表述通俗易懂，内容新颖实用，尽量用实例来诠释概念和方法，使读者能够轻松地掌握面向对象软件工程的方法和技能，进而在软件企业很快地进入各种角色。

② 突出“规范化”，让学生接受专业的课程训练。

③ 突出行业背景，让学生积累丰富的项目经验。

④ 教学材料：本书配有教学PPT和练习答案，以方便广大教师和学生使用。获得这些资料可以和出版社或作者本人联系。

四、作者分工

本书由张泊平负责设计、拟纲、统稿、定稿。全书14章的具体分工是：张泊平编写第1章、第4章；王爽编写第2章；平源编写第3章、第7章第5节；冯朝一编写第5章、第7章第1节；张琳编写第6章、第7章第3节；宋永志编写第8章、第13章；张翠善编写第9章；李慧娜编写第7章第6节、第10章；韩松编写第11章；李国庆编写第7章第4节、第12章；赵会洋编写第7章第2节、第14章；图表由城环学院吴国玺老师设计并清绘。

由于软件工程所涉及的知识面广、内容深，加上时间仓促，作者水平有限，书中的不足之处在所难免，恳请读者批评指正。在阅读本书时，读者如有意见或建议，可以发送E-mail至iebopingzhang@163.com。

编　者

2009年7月

目　录

第1章　软件工程概述 …… (1)
1.1　软件与软件危机 …… (1)
1.1.1　软件的定义、特点、种类及发展 …… (1)
1.1.2　软件危机 …… (4)
1.2　软件工程的概念 …… (5)
1.2.1　软件工程的定义 …… (6)
1.2.2　软件工程研究的内容 …… (6)
1.2.3　软件工程的基本原理 …… (7)
1.3　软件的开发方法 …… (7)
1.3.1　结构化方法 …… (7)
1.3.2　面向数据结构的方法 …… (8)
1.3.3　面向对象的开发方法 …… (8)
1.3.4　软件开发新方法 …… (9)
1.4　软件生存周期 …… (10)
1.5　软件生存周期模型 …… (11)
1.5.1　瀑布模型 …… (11)
1.5.2　原型模型 …… (12)
1.5.3　增量模型 …… (13)
1.5.4　螺旋模型 …… (14)
1.5.5　喷泉模型 …… (17)
1.5.6　智能模型 …… (18)
1.5.7　构件组装模型 …… (19)
小结 …… (20)
习题 …… (20)
第2章　可行性分析 …… (22)
2.1　可行性研究的任务 …… (22)
2.2　可行性研究的具体步骤 …… (25)
2.3　可行性研究的文档 …… (26)
2.4　项目开发计划 …… (27)
小结 …… (27)
习题 …… (27)
第3章　软件需求分析 …… (29)
3.1　软件需求分析的基本概念 …… (29)
3.1.1　软件需求分析的任务 …… (29)

3.1.2 需求分析的过程 …… (30)
3.1.3 需求分析的困难 …… (31)
3.2 分析建模 …… (31)
3.2.1 应该分析什么 …… (31)
3.2.2 通过什么方式去分析 …… (32)
3.2.3 需求分析方法 …… (33)
3.3 结构化分析方法 …… (33)
3.4 数据流图 …… (34)
3.5 数据词典 …… (40)
3.6 加工逻辑说明 …… (41)
3.7 系统行为建模 …… (43)
3.7.1 状态图 …… (43)
3.7.2 Petri 网 …… (44)
3.8 原型化分析方法 …… (45)
3.8.1 原型化方法的基本思想 …… (46)
3.8.2 原型化方法和工具 …… (46)
3.8.3 采用原型化方法的步骤 …… (46)
3.9 需求分析文档 …… (47)
3.10 案例分析——图书馆管理系统 …… (49)
3.10.1 问题陈述 …… (49)
3.10.2 图书馆组织结构 …… (50)
3.10.3 系统业务流程分析 …… (50)
3.10.4 数据流程图 …… (51)
3.10.5 数据定义及数据词典 …… (53)
3.10.6 细化需求规格说明 …… (56)
小结 …… (57)
习题 …… (57)
第4章 软件外部设计——交互设计 …… (60)
4.1 交互设计概述 …… (60)
4.1.1 交互设计的概念 …… (60)
4.1.2 交互设计与界面设计 …… (61)
4.1.3 交互设计的特征 …… (63)
4.1.4 交互设计的内容 …… (63)
4.1.5 怎样进行交互设计 …… (64)
4.1.6 如何粗略地评估可用性 …… (65)
4.2 交互设计的方法 …… (65)
4.2.1 角色设计 …… (65)
4.2.2 目标设计 …… (66)
4.2.3 任务设计 …… (67)

4.2.4 交互样式 …… (68)
4.3 案例分析——LMS 的交互设计 …… (75)
小结 …… (77)
习题 …… (77)
第 5 章 软件内部设计 …… (79)
5.1 设计过程 …… (79)
5.1.1 结构化设计与结构化分析的关系 …… (79)
5.1.2 设计和软件质量的关系 …… (80)
5.2 设计概念 …… (80)
5.2.1 软件设计的观点 …… (80)
5.2.2 控制层次 …… (83)
5.3 有效的模块设计 …… (85)
5.3.1 功能独立性 …… (85)
5.3.2 内聚性 …… (85)
5.3.3 耦合 …… (87)
5.3.4 控制范围与作用范围之间的约束 …… (88)
5.4 结构化设计方法 …… (89)
5.4.1 结构化开发方法 …… (89)
5.4.2 数据流图的分类与典型的系统结构 …… (90)
5.4.3 变换型系统结构图 …… (91)
5.4.4 事务型系统结构图 …… (93)
5.5 详细设计描述工具 …… (95)
5.5.1 结构化程序设计 …… (95)
5.5.2 图形设计符号 …… (96)
5.6 编写软件设计文档 …… (98)
5.7 案例分析——LMS 系统设计 …… (100)
5.7.1 系统功能设计 …… (100)
5.7.2 系统环境设计 …… (101)
5.7.3 数据库设计 …… (102)
小结 …… (105)
习题 …… (105)
第 6 章 面向对象软件工程基础 …… (107)
6.1 面向对象的开发方法概述 …… (107)
6.1.1 类与对象 …… (107)
6.1.2 继承 …… (108)
6.1.3 虚函数和多态性 …… (109)
6.1.4 消息 …… (111)
6.1.5 方法 …… (112)
6.2 面向对象开发模型 …… (112)

6.2.1 面向对象开发模型概述 …… (112)
6.2.2 建立模型的作用 …… (113)
6.2.3 创建优质模块 …… (113)
6.3 面向对象分析 …… (115)
6.3.1 SA 方法和 OOA 方法比较 …… (115)
6.3.2 面向对象分析的特点 …… (115)
6.3.3 面向对象分析的基本任务 …… (116)
6.3.4 OOA 过程 …… (117)
6.4 面向对象设计 …… (119)
6.4.1 面向对象设计的模型 …… (119)
6.4.2 面向对象设计的任务 …… (123)
6.4.3 优质对象系统的属性 …… (125)
6.5 案例分析——LMS 中的产品设计 …… (126)
6.5.1 面向对象概念化 …… (126)
6.5.2 耦合 …… (127)
6.5.3 确定系统的参与者 …… (128)
小结 …… (128)
习题 …… (128)
第 7 章 统一建模语言 …… (131)
7.1 UML 概述 …… (131)
7.1.1 UML 的发展 …… (131)
7.1.2 UML 的构成 …… (132)
7.1.3 视图 …… (132)
7.1.4 UML 的图形表示 …… (134)
7.1.5 UML 的通用模型元素 …… (135)
7.2 用例模型 …… (139)
7.2.1 用例图 …… (139)
7.2.2 参与者 …… (139)
7.2.3 用例 …… (141)
7.3 建立静态模型 …… (145)
7.3.1 类图 …… (145)
7.3.2 对象图 …… (151)
7.3.3 包图 …… (152)
7.4 动态模型 …… (154)
7.4.1 对象之间的交互——消息 …… (154)
7.4.2 状态图 …… (155)
7.4.3 时序图 …… (158)
7.4.4 协作图 …… (162)
7.4.5 活动图 …… (164)

7.5 实现模型 …… (166)
7.5.1 构件图 …… (166)
7.5.2 配置图 …… (168)
7.6 案例分析——LMS 的 UML 模型 …… (169)
7.6.1 UCCD 类的分析过程 …… (169)
7.6.2 LMS 分析 …… (172)
7.6.3 LMS 设计 …… (178)
7.6.4 LMS 的交互图 …… (183)
小结 …… (187)
习题 …… (187)
第 8 章 软件编码与实现 …… (189)
8.1 程序设计语言 …… (189)
8.1.1 程序设计语言的分类 …… (189)
8.1.2 程序设计语言的选择 …… (190)
8.2 程序设计基础 …… (191)
8.2.1 结构化程序设计 …… (191)
8.2.2 程序设计风格 …… (191)
8.2.3 程序效率 …… (194)
8.3 编程安全 …… (195)
8.3.1 保护性编程 …… (195)
8.3.2 冗余编程 …… (196)
8.4 软件实现后编写的文档 …… (196)
8.5 案例分析——LMS 系统实现（编码） …… (198)
8.5.1 读者注册系统的实现 …… (198)
8.5.2 读者留言板的实现 …… (200)
小结 …… (203)
习题 …… (203)
第 9 章 软件测试 …… (204)
9.1 软件测试的基本概念 …… (204)
9.1.1 软件测试的目的和重要性 …… (204)
9.1.2 软件测试的原则 …… (206)
9.1.3 软件测试的几个定义 …… (207)
9.1.4 软件测试的基本步骤 …… (209)
9.1.5 静态分析与动态测试 …… (209)
9.2 软件测试的基本方法 …… (210)
9.2.1 测试用例 …… (210)
9.2.2 白盒测试 …… (211)
9.2.3 黑盒测试 …… (214)
9.3 软件测试的策略 …… (217)

9.3.1 单元测试 …… (218)
9.3.2 组装测试 …… (219)
9.3.3 确认测试 …… (222)
9.3.4 系统测试 …… (223)
9.4 排错技术 …… (224)
9.4.1 排错的原则 …… (225)
9.4.2 排错方法 …… (226)
9.5 面向对象的测试 …… (227)
9.5.1 面向对象测试原理 …… (228)
9.5.2 面向对象的单元测试 …… (228)
9.5.3 面向对象的集成测试 …… (229)
9.5.4 面向对象的确认测试 …… (230)
9.6 软件测试计划与测试分析报告 …… (230)
9.6.1 软件测试计划 …… (230)
9.6.2 测试分析报告 …… (233)
9.7 LMS 测试 …… (235)
9.7.1 测试计划 …… (235)
9.7.2 系统地提出测试用例 …… (237)
小结 …… (238)
习题 …… (239)
第 10 章 软件维护 …… (241)
10.1 软件维护的定义、分类、特点 …… (241)
10.1.1 软件维护的定义 …… (241)
10.1.2 软件维护的分类 …… (242)
10.1.3 软件维护的特点 …… (243)
10.2 软件维护过程及组织 …… (244)
10.2.1 软件维护过程 …… (244)
10.2.2 软件维护组织 …… (245)
10.3 软件的可维护性 …… (246)
10.3.1 软件的可维护性概念 …… (246)
10.3.2 影响可维护性的因素 …… (247)
10.3.3 提高软件可维护性的方法 …… (250)
10.3.4 软件维护的副作用 …… (253)
小结 …… (254)
习题 …… (254)
第 11 章 软件复用和构件技术 …… (256)
11.1 软件复用概述 …… (256)
11.1.1 软件复用的概念 …… (256)
11.1.2 软件复用的分类和复用级别 …… (256)

11.1.3 软件复用的形式 …… (258)
11.1.4 软件复用的实施过程 …… (258)
11.2 可复用构件与构件工程 …… (259)
11.2.1 可复用构件 …… (259)
11.2.2 基于构件的软件工程 …… (260)
11.3 领域分析和基于构件的开发 …… (261)
11.3.1 领域分析 …… (261)
11.3.2 构件的开发与构件库 …… (261)
11.4 基于构件的开发 …… (264)
小结 …… (265)
习题 …… (266)
第12章 软件项目计划与管理 …… (267)
12.1 软件项目管理 …… (267)
12.1.1 软件项目管理的特点 …… (267)
12.1.2 软件工程管理的内容 …… (267)
12.2 软件项目计划 …… (269)
12.2.1 软件开发进度计划 …… (269)
12.2.2 软件项目计划内容 …… (269)
12.2.3 软件项目进度安排 …… (270)
12.3 项目成本估算 …… (274)
12.3.1 影响成本估算的因素 …… (274)
12.3.2 成本估算模型 …… (275)
12.3.3 Halstead 理论模型 …… (275)
12.3.4 专家估算模型 …… (275)
12.3.5 IBM 估算模型 …… (276)
12.3.6 Putnam 估算模型 …… (276)
12.3.7 COCOMO 模型 …… (277)
12.3.8 成本估算方法 …… (279)
12.4 软件复杂性 …… (279)
12.4.1 软件复杂性的基本概念 …… (279)
12.4.2 软件复杂性的度量方法 …… (280)
12.5 软件可靠性 …… (281)
12.5.1 软件可靠性定义 …… (281)
12.5.2 软件可靠性指标 …… (281)
12.5.3 软件可靠性模型 …… (282)
12.6 软件能力成熟度模型（CMM） …… (283)
12.6.1 CMM 的基本概念 …… (283)
12.6.2 软件过程的成熟度等级 …… (284)
12.6.3 在软件企业中实施 CMM …… (285)

小结 …… (287)
习题 …… (287)
第 13 章 软件工程环境 …… (289)
13.1 软件工程环境概述 …… (289)
13.1.1 软件工程环境定义 …… (289)
13.1.2 软件开发环境的特性 …… (289)
13.1.3 软件开发环境的分类 …… (290)
13.2 软件开发工具 …… (291)
13.3 软件开发工具实例 Rational Rose …… (293)
13.3.1 Rose 工具简介 …… (293)
13.3.2 业务用例图 …… (294)
13.3.3 用例图 …… (295)
13.3.4 类图 …… (296)
13.3.5 协作图与时序图 …… (298)
13.3.6 活动图 …… (299)
13.3.7 状态图 …… (300)
13.3.8 构件图和部署图 …… (301)
13.3.9 正向工程和逆向工程 …… (302)
小结 …… (304)
习题 …… (304)
第 14 章 软件工程课程设计 …… (305)
14.1 课程简介 …… (305)
14.2 课程设计参考案例——音像社信息管理系统 …… (306)
14.2.1 开发前的准备 …… (306)
14.2.2 描述企业概图 …… (308)
14.2.3 系统分析 …… (309)
14.2.4 系统设计 …… (312)
14.2.5 系统实施 …… (315)
14.2.6 系统运行与维护 …… (317)
14.3 汽车租赁企业信息管理系统 …… (317)
14.3.1 汽车租赁系统的需求分析 …… (317)
14.3.2 系统的时序图 …… (322)
14.3.3 系统的协作图 …… (325)
14.3.4 系统的状态图 …… (326)
14.3.5 系统的活动图 …… (327)
14.3.6 系统中的类 …… (328)
14.3.7 系统的配置模型与实现模型 …… (330)
14.3.8 系统实现 …… (332)
参考文献 …… (335)
后记 …… (336)

第1章　软件工程概述

教学提示： 本章介绍软件工程和软件开发方法的有关知识，主要包括软件危机的概念、软件工程的概念、常用的软件开发方法、软件生存周期模型等4个方面的内容。

教学目标： 了解软件危机的特点、常用的软件开发工具，掌握软件工程的概念、常用的软件开发方法和几种基本的软件生存期模型，掌握与上述内容相关的知识。

随着计算机技术的飞速发展、网络技术的普及，当今社会已经进入了以计算机为核心的信息社会。在信息社会中，信息的获取、处理和交流等都需要高质量的软件产品。若想使软件功能强，使用方便，开发出来的软件产品就会复杂和庞大，开发人员的能力显得力不从心，以致软件开发计划不能按时进行，成本失去控制，软件质量得不到保证等，从而导致软件危机。为了克服这种现象，自20世纪60年代末期以来，人们十分重视软件开发过程的研究，包括开发方法、开发工具和开发环境等，同时在这些领域取得了重大成果，从而产生了软件工程理论。

1.1　软件与软件危机

本节讨论软件的定义，与硬件对比归纳了软件的特点，按3种方式对软件进行分类，并介绍了软件的发展与软件危机。

1.1.1　软件的定义、特点、种类及发展

1. 软件的定义

二十多年以来，计算机软件的发展速度是十分惊人的：在体系结构方面，经历了从主机结构到文件服务器结构、从客户－服务器(C/S)结构到基于Internet的浏览器－服务器(B/S)结构等变化；在编码语言方面，经历了从机器代码到汇编代码、从高级程序语言到人工智能语言等变化；在开发工具方面，经历了从分离的开发工具到集成的可视化开发系统、从简单的命令行调试器到方便的多功能调试器等变化。

与软件的发展速度相比，软件的基本定义在过去的几十年当中并未发生太大的变化。可能有人认为软件就是程序，这个理解是片面的。1983年，IEEE(电气电子工程师协会)为计算机软件下的定义是：计算机程序、方法、规则和相关的文档资料及在计算机上运行时所必需的数据。目前对计算机软件通俗的理解为：包括程序、数据及其相关文档资料的完整集合，即

$$\text{软件} = \text{程序} + \text{数据} + \text{文档资料}$$

其中，程序是完成特定功能和满足性能要求的指令序列；数据是程序运行的基础和操作的对象；文档资料是与程序开发、维护和使用有关的图文资料。

2. 软件的特点

与硬件产品相比，软件的特点可归纳如下。

(1) 软件是一种逻辑产品

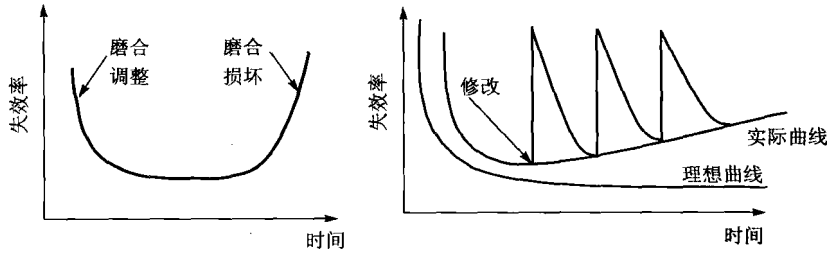
软件是一种逻辑产品,而不是具体的物理实体,因而具有抽象性。人们可以把它记录在介质上,却无法看到软件的形态,必须通过测试、分析、思考、判断去了解它的功能及其特性。

(2) 软件没有明显的制造过程

软件的生产与硬件不同,没有明显的制造过程。其生产过程主要表现为人脑的思维过程,具有不可见性,并且带有个人色彩。

(3) 软件不存在机械磨损或老化问题

任何机械、电子设备在运行和使用中,其失效率大都遵循图 1-1(a)所示的 U 形曲线(浴盆曲线)。而软件的情况与此不同,它不存在机械磨损和老化问题,而存在退化问题。为了适应硬件、系统环境及需求的变化,必须多次修改(维护)软件。软件的修改不可避免地会引入新的错误,导致软件的失效率升高,使得软件的可靠性下降,如图 1-1(b)所示。当修改的成本变得难以接受时,软件就会被抛弃。



(a) 硬件失效率曲线　　(b) 软件失效率曲线

图 1-1　失效率

(4) 软件对硬件和环境的依赖性

软件对硬件和环境有着不同程度的依赖性,这导致了软件升级和移植的问题。随着计算机硬件和支撑环境的不断升级,计算机软件也需要不断维护,并且维护成本通常比开发成本高许多。

(5) 软件的复杂性越来越高

随着软件需求的增长,软件所处理的对象类型由单纯的数值型发展到字符、图形、声音等类型,软件处理问题的规模日趋庞大。IBM360 操作系统第 16 版有 100 万条指令,1973 年美国阿波罗计划达 1 000 万条指令。这些庞大软件的程序逻辑结构、调用关系、接口信息及数据结构等都很复杂。随着软件规模的增大,对软件人员的要求也越来越高,其复杂程度超出了人们能接受的程度,因此出现了软件复杂性与软件技术发展的不适应现象,如图 1-2 所示。

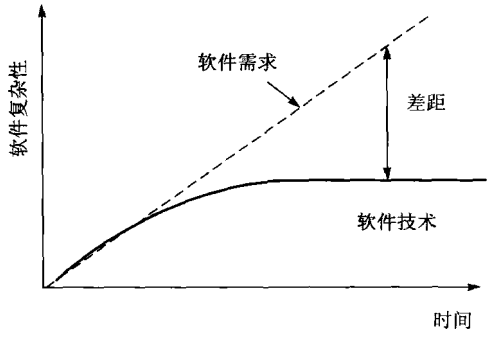


图 1-2　软件技术进步落后于需求复杂性的增长

(6) 软件成本昂贵

软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，它的成本自 20 世纪 80 年代以来已大大超过硬件成本，硬件/软件成本比率的变化趋势如图 1 - 3 所示。

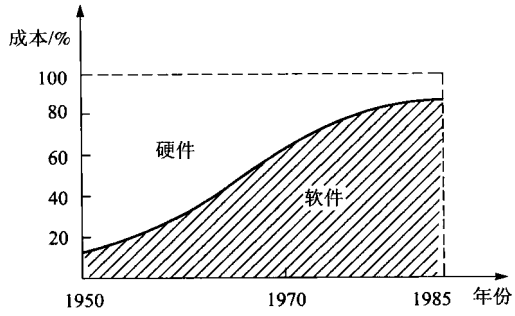


图 1 - 3　硬件/软件成本比率的变化趋势

(7) 软件开发工作涉及许多社会因素

许多社会因素，如机构、体制、管理方式等，包括人的观念及心理，都直接影响到软件工作的成败。

(8) 软件开发工作任重道远

大多数软件是新开发的，而不是通过已有的构件组装而来的。硬件的设计和建造可通过画电路图、做一些基本的分析以保证实现预定的功能，根据功能和接口要求选定并购买零件；而软件设计中几乎没有软件构件。虽然关于“软件复用”已有大量论著，但这个概念的成功实现还有很长的路要走。

以上特点使得软件开发进展情况较难衡量，软件开发质量难以评价，从而使得软件产品的生产管理、过程控制及质量保证都相当困难。

3. 软件的种类

随着软件技术的不断发展，支持人们日常学习、工作的软件产品的种类和数量都已经很多。由于人们对软件关心的侧重点不同，对软件的分类也很难有一个科学、统一的标准。

但对软件的类型进行必要的划分，根据不同类型的工程对象采用不同的开发和维护方法是很有价值的，因此有必要从不同的角度讨论计算机软件的分类情况。

(1) 按软件的功能分类

可分为系统软件、支撑软件和应用软件 3 类。

(2) 按软件规模分类

即按照开发软件所需的人力、物力、时间及完成的源程序行数进行分类，可分为 6 种。

① 微型软件：一个人在几天之内可以完成的、程序语句不超过 500 行的程序。

② 小型软件：一个人在半年之内可以完成的 2 000 行以内的程序。

③ 中型软件：不超过 5 个人在 2 年内能够完成的 5 000 ～ 50 000 行的程序。

④ 大型软件：5 ～ 20 人在 2 ～ 3 年的时间里完成的50 000～ 100 000 行的程序。

⑤ 甚大型软件：由 100 ～ 1 000 人参加，用 4 ～ 5 年时间完成的具有 100 万行程序的软件项目。

⑥ 极大型软件：由 2 000 ～ 5 000 人参加，10 年内能够完成的 1 000 万行以内的程序。

(3) 按软件工作方式分类

可分为4类:实时处理软件、分时软件、交互式软件、批处理软件。

4. 软件的发展

自从第一台计算机诞生以来,就开始了软件的生产,到目前为止,软件发展经历了4个阶段。

(1) 程序设计阶段

早期阶段(20世纪50年代初期至60年代中期)为程序设计阶段。在这个阶段硬件已经通用化,而软件的生产却是个体化的。由于程序规模小,几乎没有系统化的方法可以遵循。对软件的开发没有任何管理方法,在出现计划推迟或者成本提高的情况时,程序员才开始弥补。软件产品还处在初级阶段,对每一类应用均需自行再设计,软件应用范围很有限。软件设计往往仅是人们头脑中的一种模糊想法,而文档根本不存在。

(2) 程序系统阶段

第二阶段(20世纪60年代中期到70年代末期)为程序系统阶段。多道程序设计、多用户系统引入了人机交互的概念。交互技术打开了计算机应用的新领域,使硬件和软件的配合达到了一个新的层次,实时系统和第一代数据库管理系统出现了。这个阶段的另一个特点是软件产品的使用和"软件作坊"的出现。被开发的软件可以在较宽广的范围内应用。主机和微机上的基础程序能够有数百甚至上千的用户。

在软件的使用过程中,当发现程序错误或用户需求和硬件环境发生变化时,都需要修改软件,这些活动统称为软件维护。在软件维护上的花费以惊人的速度增长。更为严重的是,许多程序的个体化特性使得它们根本不能维护。"软件危机"出现了。

(3) 软件工程阶段

第三阶段始于20世纪70年代中期并历时近十年,被称为软件工程阶段。在这一阶段,以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来,打破了软件生产的个体化特征,有了可以遵循的软件工程化的设计原则、方法和标准。在分布式系统中,各台计算机同时执行某些功能,并与其他计算机通信,极大地提高了计算机系统的复杂性。广域网、局域网、高带宽数字通信及对"即时"数据访问需求的增加都对软件开发者提出了更高的要求。

(4) 第四阶段

第四阶段已经不再着重于单台计算机和计算机程序,而是针对计算机和软件的综合影响。由复杂操作系统控制的强大的台式机、广域和局域网络,配以先进的软件应用已成为标准。计算机体系结构迅速地从集中的主机环境转变为分布的客户-服务器环境。世界范围的信息网提供了一个基本结构,信息高速公路和网际空间的联通已成为令人关注的热点问题。事实上,Internet可以看作是能够被单个用户访问的软件,计算机朝着社会信息化和软件产业化的方向发展,从技术的软件工程阶段过渡到社会信息化的计算机系统。随着第四阶段的进展,一些新技术开始涌现。面向对象技术在许多领域中迅速取代传统的软件开发方法。

1.1.2 软件危机

软件危机是指在软件开发和维护过程中遇到的一系列严重问题。这些问题绝不仅是"不能正常运行的"软件才具有的,实际上几乎所有软件都不同程度地存在这些问题。概括地说,软件危机包含两方面的问题:① 如何开发软件,怎样满足对软件的日益增长的需求;② 如何

维护数量不断膨胀的已有软件。

1. 软件危机的主要表现

具体地说，软件危机主要有以下表现。

① 产品不符合用户的实际需要。

② 软件开发生产率提高的速度远远不能满足客观需要，软件的生产率远远低于硬件生产率和计算机应用需求的增长，使人们不能充分利用现代计算机硬件提供的巨大潜力。

③ 软件产品的质量差。软件可靠性和质量保证的定量概念刚刚出现不久，软件质量保证技术（审查、复审和测试）没有贯穿到软件开发的全过程中，这些都导致软件产品发生质量问题。

④ 对软件开发成本和进度的估计常常不准确。实际成本比估计成本有可能高出一个数量级，实际进度比预期进度拖延几个月甚至几年。这种现象损害了软件开发者的信誉，而为了赶进度和节约成本所采取的一些权宜之计又往往降低了软件产品的质量，从而不可避免地引起用户的不满。

⑤ 软件的可维护性差。很多程序中的错误是难以改正的，实际上不能使这些程序适应硬件环境的改变，也不能根据用户的需要在原有程序中增加一些新的功能。没能实现软件的可重用，人们仍然在重复开发功能类似的软件。

⑥ 软件文档资料通常既不完整又不合格。

⑦ 软件的价格昂贵。软件成本在计算机系统总成本中所占的比例逐年上升。

2. 产生软件危机的原因

① 缺乏总体考虑，没有软件工程学概念或系统工程思想。

② 对业务的了解支离破碎，需求分析不准。

③ 企业依赖激情指挥，企业管理的标准化、规范化、科学化程度不高，导致不能成功地应用“死板”的软件，它依赖于业务的“科学化”、“条理化”、“程序化”。

④ 企业信息化程度和计算机应用水平低，导致无法准确描述需求。

⑤ 企业的高层管理人员对信息管理的重视程度不够。

⑥ 缺乏相互沟通，业务描述的详尽程度不能达到使具备业务常识的人能够轻易理解。

3. 软件危机解决途径

① 软件开发应该是一种组织良好、管理严密、各类人员协同配合、共同完成的生产项目，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概念、原理、技术和方法。

② 应该推广和使用在软件开发实践中总结出来的成功的技术和方法，并且研究探索更好、更有效的技术和方法，尽快纠正在计算机早期发展阶段形成的错误概念和做法。

③ 应该制定软件开发过程的规范和标准。因为人类的一切生产活动只有纳入科学的、规范的轨道，执行有约束力的标准，才是最高效的。

④ 应该开发和使用更多、更好的软件工具（Software Tools），借鉴“利用机械工具可以放大人的体力”的原理，软件工具可以“放大”人的智力。

1.2　软件工程的概念

本节讨论软件工程的定义，并介绍软件工程的研究内容、基本目标及开发原则。