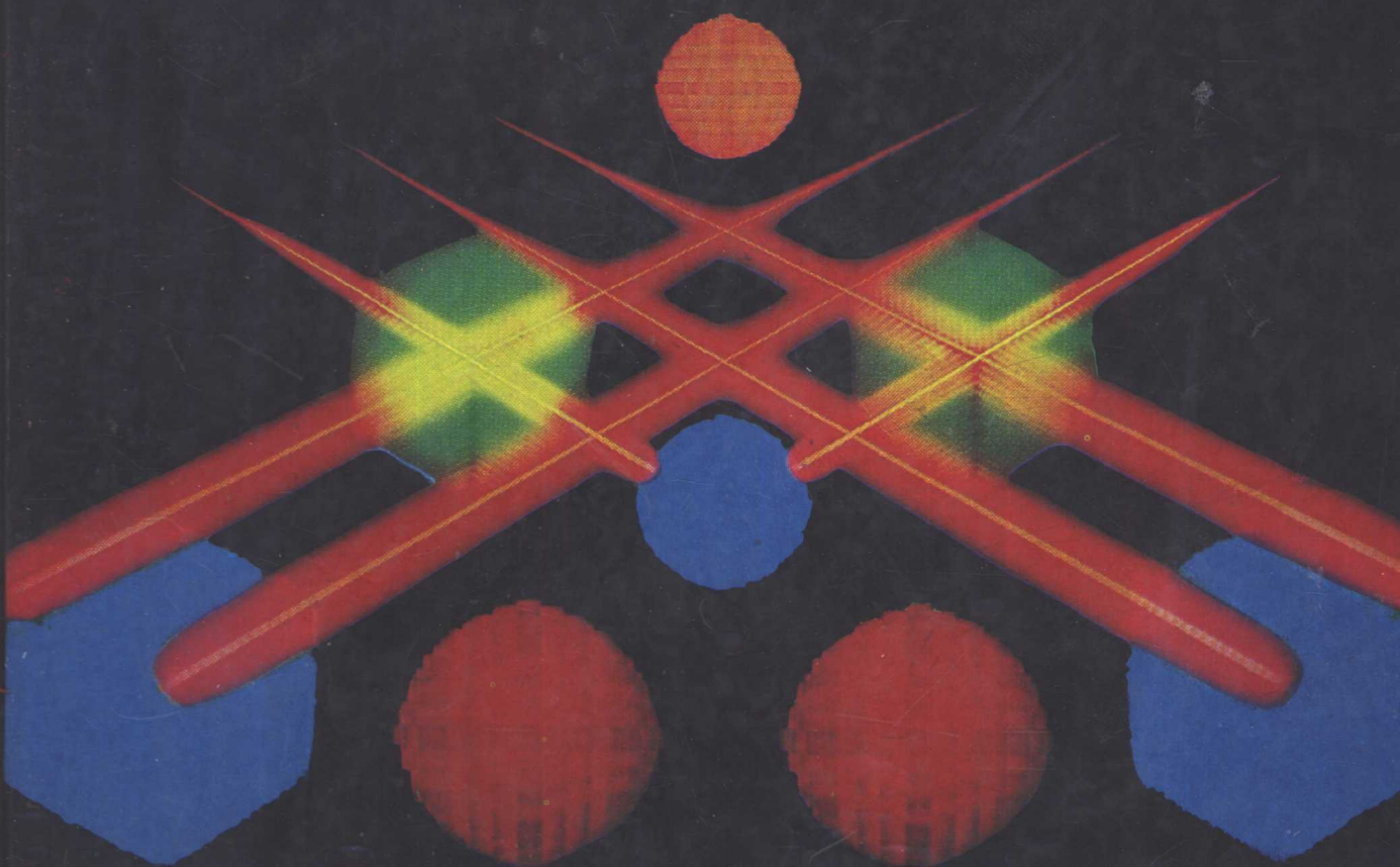


# 软件工程 及其应用

周苏 陈敏玲 陈根才 王文 等编著



天津科学技术出版社

Tip  
8700

# 软件工程及其应用

周 苏 陈敏玲 等编著  
陈根才 王 文

天津科学技术出版社

津新登字(90)003号

责任编辑:徐 彤

软件工程及其应用

周 苏 陈敏玲 等编著  
陈根才 王 文

\*

天津科学技术出版社出版

天津市赤峰道130号

天津新华印刷二厂印刷

新华书店天津发行所发行

\*

开本 787×1092毫米 1/16 印张 26.75·字数: 840 000

1992年9月第1版

1992年9月第1次印刷

印数: 1—3 000

ISBN 7-5308-1094-4/TP·31 定价: 19.45元

## 内 容 提 要

本书系统、全面地介绍了有关“软件工程”的概念、方法、工具及其应用，并且力图反映国内外软件工程领域的最新成果。全书分软件工程概述、系统定义与软件计划、软件需求分析、概要设计、详细设计、软件界面设计、数据库／数据结构设计、软件安全性设计、软件编码、软件测试、软件文件、软件维护、软件工具和软件开发环境、软件的质量与评价、软件管理等共十五章和一个附录。书中选编了大量的习题和思考题。

本书可作为高等学校“软件工程”课程的教材和教学参考书，可供有一定实践经验的软件开发人员、管理人员阅读和作继续教育的教材，也非常适合用作各个级别（尤其是高级程序员、系统分析员）的计算机软件专业技术资格和水平考试有关内容的学习辅导教材。

# 序

软件工程是人们用以指导软件开发的一门新技术。软件工程学主要研究软件结构、软件设计方法、软件工具、软件工程标准和规范以及软件工程的有关理论。

本书系统、全面地介绍了有关“软件工程”的技术，在软件工程的观念、原理、方法及其应用等方面有较详尽的论述，有较强的系统性和可读性，并力图反映国内外软件工程领域的最新成果。全书分软件工程概述、系统定义与软件计划、软件需求分析、概要设计、详细设计、软件界面设计、数据库 / 数据结构设计、软件安全性设计、软件编码、软件测试、软件文件、软件维护、软件工具和软件开发环境、软件的质量与评价、软件管理等共十五章和一个附录。书中选编了大量的习题和思考题，可作为高等学校“软件工程”课程的教材和教学参考书。

按照国家标准局有关软件工程的各个标准来表达和描述软件工程的知识，是本书的特色之一。GB8566-88《计算机软件开发规范》、GB8567-88《计算机软件产品开发文件编制指南》等国家标准的有关要求，贯穿于本书的始终。

用软件工程方法来指导应用软件的开发实践，是本书的另一个特色。书中包含有许多应用实例，并结合应用实践对软件开发方法进行论述，本书可供有一定实际经验的软件开发人员、管理人员阅读和作继续教育的教材。

本书涉及面广，内容翔实，可用作程序员、高级程序员和系统分析员等各个级别的软件人员水平考试中，有关软件设计能力、软件工程知识等内容的学习辅导用书。

希望本书能适应不同层次读者的需要，受到广大读者的青睐。同时，我们也恳请广大读者能给予指教，以期再版时得以补正。

参加本书编写的还有：王大春、吕茜、朱琪、赵迅、童志强、柳俊等。

浙江省软件技术开发中心副总工程师、副研究员孙伟先生对本书作了认真、细致的审阅，并提出了宝贵的修改意见。

我们衷心感谢在本书的编写过程中，直接、间接地给予了很大帮助的许多专家、学者。

编 者

1992 年 1 月

# 目 录

|                         |        |
|-------------------------|--------|
| 第一章 软件工程概述 .....        | ( 1 )  |
| 1.1 计算机系统工程 .....       | ( 1 )  |
| 1.1.1 计算机系统的发展 .....    | ( 1 )  |
| 1.1.2 计算机系统工程 .....     | ( 2 )  |
| 1.2 软件和软件生存周期 .....     | ( 3 )  |
| 1.3 软件工程 .....          | ( 5 )  |
| 1.4 软件工程学的基本原则 .....    | ( 7 )  |
| 1.5 软件工程的进展 .....       | ( 8 )  |
| 1.6 小结 .....            | ( 9 )  |
| 习题与思考 .....             | ( 9 )  |
| <br>第二章 系统定义与软件计划 ..... | ( 11 ) |
| 2.1 系统定义 .....          | ( 11 ) |
| 2.1.1 系统分析 .....        | ( 12 ) |
| 2.1.2 可行性研究 .....       | ( 16 ) |
| 2.1.3 成本-收益分析 .....     | ( 17 ) |
| 2.1.4 功能分配 .....        | ( 19 ) |
| 2.1.5 系统规格说明 .....      | ( 19 ) |
| 2.1.6 系统定义复审 .....      | ( 20 ) |
| 2.2 软件计划 .....          | ( 21 ) |
| 2.2.1 软件的范围 .....       | ( 21 ) |
| 2.2.2 资源 .....          | ( 22 ) |
| 2.2.3 软件成本估算 .....      | ( 24 ) |
| 2.3 成本估算方法 .....        | ( 24 ) |
| 2.3.1 软件生产率数据 .....     | ( 24 ) |
| 2.3.2 估算模型 .....        | ( 26 ) |
| 2.3.3 语句行成本估算方法 .....   | ( 27 ) |
| 2.3.4 工作量成本估算方法 .....   | ( 29 ) |
| 2.3.5 自动化成本估算技术 .....   | ( 30 ) |
| 2.4 进度安排 .....          | ( 30 ) |
| 2.5 软件计划文件及复审 .....     | ( 32 ) |
| 2.6 小结 .....            | ( 34 ) |
| 习题与思考 .....             | ( 34 ) |

|                         |      |
|-------------------------|------|
| <b>第三章 软件需求分析</b>       | (36) |
| 3.1 需求分析阶段的任务           | (36) |
| 3.2 结构化分析方法 (SA)        | (37) |
| 3.3 数据流程图               | (39) |
| 3.3.1 数据流程图的准则          | (42) |
| 3.4 数据字典                | (43) |
| 3.5 加工的分析与表达            | (46) |
| 3.5.1 结构化语言             | (47) |
| 3.5.2 判定表               | (48) |
| 3.5.3 判定树               | (50) |
| 3.5.4 三种表达工具的比较         | (50) |
| 3.6 需求分析工具              | (51) |
| 3.6.1 结构化分析和设计技术 (SADT) | (51) |
| 3.6.2 SREM 工具           | (52) |
| 3.6.3 PSL/PSA 工具        | (53) |
| 3.7 软件需求分析文件与复审         | (53) |
| 3.8 小结                  | (59) |
| 习题与思考                   | (59) |
| <b>第四章 概要设计</b>         | (63) |
| 4.1 模块的划分               | (63) |
| 4.1.1 系统性能的衡量标准         | (63) |
| 4.1.2 软件结构              | (64) |
| 4.1.3 模块划分的基本原则         | (65) |
| 4.1.4 模块划分的方法           | (65) |
| 4.2 结构化设计方法 (SD)        | (69) |
| 4.2.1 变换与事务型数据流分析       | (69) |
| 4.2.2 模块化设计             | (72) |
| 4.2.3 模块结构图 (SC)        | (73) |
| 4.2.4 从数据流程图导出模块结构图     | (74) |
| 4.3 Parnas 方法           | (76) |
| 4.4 Jackson 方法          | (77) |
| 4.5 程序的逻辑构造 (LCP) 方法    | (79) |
| 4.6 概要设计文件与复审           | (80) |
| 4.7 小结                  | (81) |
| 习题与思考                   | (81) |
| <b>第五章 详细设计</b>         | (87) |
| 5.1 详细设计概述              | (87) |

|                                 |             |
|---------------------------------|-------------|
| 5.2 结构化构造 .....                 | ( 88 )      |
| 5.3 图形设计工具 .....                | ( 89 )      |
| 5.3.1 流程图 .....                 | ( 89 )      |
| 5.3.2 方块图 (N-S) .....           | ( 92 )      |
| 5.3.3 HIPO 图 .....              | ( 94 )      |
| 5.3.4 PAD 图 .....               | ( 95 )      |
| 5.4 伪码与程序设计语言 (PDL) .....       | ( 98 )      |
| 5.5 各种详细设计工具的比较 .....           | ( 100 )     |
| 5.6 详细设计文件与复审 .....             | ( 102 )     |
| 5.7 小结 .....                    | ( 103 )     |
| 习题与思考 .....                     | ( 103 )     |
| <br><b>第六章 软件界面设计</b> .....     | <br>( 106 ) |
| 6.1 代码设计 .....                  | ( 106 )     |
| 6.1.1 代码的基础知识 .....             | ( 106 )     |
| 6.1.2 代码的种类 .....               | ( 107 )     |
| 6.1.3 各种代码形态的比较 .....           | ( 109 )     |
| 6.1.4 代码设计步骤 .....              | ( 110 )     |
| 6.1.5 代码设计文件 .....              | ( 110 )     |
| 6.2 输出设计 .....                  | ( 111 )     |
| 6.2.1 输出设计的研究范围 .....           | ( 111 )     |
| 6.2.2 输出形式的种类 .....             | ( 112 )     |
| 6.2.3 输出设计方法 .....              | ( 113 )     |
| 6.2.4 输出设计的技术和原则 .....          | ( 116 )     |
| 6.3 输入设计 .....                  | ( 118 )     |
| 6.3.1 输入设计的研究范围 .....           | ( 118 )     |
| 6.3.2 输入形式的种类 .....             | ( 119 )     |
| 6.3.3 输入设计方法 .....              | ( 119 )     |
| 6.3.4 输入设计的技术和原则 .....          | ( 123 )     |
| 6.4 用户界面的设计 .....               | ( 125 )     |
| 6.4.1 用户界面开发的基本原则 .....         | ( 125 )     |
| 6.4.2 菜单技术 .....                | ( 128 )     |
| 6.4.3 出错控制与处理 .....             | ( 130 )     |
| 6.4.4 一个出错控制与处理程序的实例 .....      | ( 131 )     |
| 6.5 小结 .....                    | ( 133 )     |
| 习题与思考 .....                     | ( 133 )     |
| <br><b>第七章 数据库/数据结构设计</b> ..... | <br>( 137 ) |
| 7.1 数据模型规范化 .....               | ( 137 )     |

|                          |              |
|--------------------------|--------------|
| 7.1.1 数据模型的构造和分类 .....   | (137)        |
| 7.1.2 关系模型的规范化 .....     | (139)        |
| 7.2 数据存储文件设计 .....       | (144)        |
| 7.2.1 文件设计内容 .....       | (144)        |
| 7.2.2 文件类别 .....         | (144)        |
| 7.2.3 文件媒体的选择 .....      | (145)        |
| 7.2.4 文件组织方式 .....       | (146)        |
| 7.2.5 文件记录格式设计 .....     | (149)        |
| 7.2.6 文件容量估算 .....       | (149)        |
| 7.3 数据库系统概述 .....        | (150)        |
| 7.4 数据库系统结构 .....        | (152)        |
| 7.5 数据库管理系统的组成 .....     | (156)        |
| 7.6 数据库的设计 .....         | (158)        |
| 7.7 实体联系法 (ER 方法) .....  | (161)        |
| 7.8 面向对象数据库 .....        | (165)        |
| 7.9 选择数据库软件 .....        | (165)        |
| 7.9.1 一般特点 .....         | (167)        |
| 7.9.2 主要系统指标 .....       | (168)        |
| 7.9.3 便利操作的特点 .....      | (169)        |
| 7.9.4 数据库语言 .....        | (170)        |
| 7.9.5 屏幕格式 .....         | (171)        |
| 7.9.6 定义报告 .....         | (172)        |
| 7.9.7 数据库软件的选择 .....     | (172)        |
| 7.10 数据库设计文件 .....       | (173)        |
| 7.11 小结 .....            | (174)        |
| 习题与思考 .....              | (174)        |
| <br>                     |              |
| <b>第八章 软件安全性设计 .....</b> | <b>(183)</b> |
| 8.1 系统安全的基本概念 .....      | (183)        |
| 8.1.1 系统安全控制的目的 .....    | (183)        |
| 8.1.2 系统中不安全的因素 .....    | (184)        |
| 8.2 软件安全控制的基本方法 .....    | (185)        |
| 8.3 软件的安全控制设计 .....      | (190)        |
| 8.3.1 初级阶段的安全控制计划 .....  | (190)        |
| 8.3.2 开发阶段的安全控制设计 .....  | (192)        |
| 8.3.3 系统运行中的安全控制 .....   | (195)        |
| 8.4 计算机病毒的防治 .....       | (196)        |
| 8.4.1 计算机病毒的特点和种类 .....  | (197)        |
| 8.4.2 计算机病毒预防与消除 .....   | (198)        |

|                             |       |
|-----------------------------|-------|
| 8.5 小结 .....                | (200) |
| 习题与思考 .....                 | (200) |
| <br>                        |       |
| <b>第九章 软件编码</b> .....       | (202) |
| 9.1 结构化程序设计方法 (SP) .....    | (202) |
| 9.2 程序设计语言 .....            | (204) |
| 9.3 程序设计风格 .....            | (208) |
| 9.3.1 源程序 .....             | (209) |
| 9.3.2 数据说明 .....            | (210) |
| 9.3.3 语句结构 .....            | (211) |
| 9.3.4 输入和输出 .....           | (211) |
| 9.4 源代码文件 .....             | (211) |
| 9.5 冗余程序设计 .....            | (214) |
| 9.6 防错性程序设计 .....           | (214) |
| 9.7 程序设计质量的评价 .....         | (215) |
| 9.8 软件编码工具 .....            | (216) |
| 9.9 面向对象的程序设计方法 .....       | (216) |
| 9.9.1 面向对象程序设计方法概述 .....    | (217) |
| 9.9.2 面向对象的程序语言 .....       | (219) |
| 9.9.3 面向对象方法与其他方法的比较 .....  | (220) |
| 9.10 系统开发的原型法与第四代语言 .....   | (221) |
| 9.10.1 原型法概述 .....          | (221) |
| 9.10.2 原型法的主要优点 .....       | (222) |
| 9.10.3 对生存周期方法的影响 .....     | (223) |
| 9.10.4 原型法软件工具——第四代语言 ..... | (223) |
| 9.10.5 第四代语言的几个问题 .....     | (224) |
| 9.11 软件编码文件与复审 .....        | (227) |
| 9.12 小结 .....               | (227) |
| 习题与思考 .....                 | (227) |
| <br>                        |       |
| <b>第十章 软件测试</b> .....       | (234) |
| 10.1 测试的基本概念 .....          | (234) |
| 10.2 测试方法 .....             | (236) |
| 10.2.1 静态分析技术 .....         | (236) |
| 10.2.2 动态测试技术 .....         | (237) |
| 10.3 单元测试 .....             | (238) |
| 10.4 组装测试 .....             | (239) |
| 10.5 确认测试 .....             | (243) |
| 10.6 测试用例设计 .....           | (244) |

|                            |           |
|----------------------------|-----------|
| 10.6.1 白盒法 .....           | (244)     |
| 10.6.2 黑盒法 .....           | (247)     |
| 10.7 测试工具 .....            | (249)     |
| 10.8 测试文件 .....            | (251)     |
| 10.8.1 测试文件综述 .....        | (251)     |
| 10.8.2 测试文件的内容要求 .....     | (252)     |
| 10.8.3 测试文件编制指南 .....      | (255)     |
| 10.9 测试的复审 .....           | (256)     |
| 10.10 排错技术 .....           | (258)     |
| 10.10.1 排错的办法 .....        | (258)     |
| 10.11 系统转换 .....           | (261)     |
| 10.12 小结 .....             | (262)     |
| 习题与思考 .....                | (262)     |
| <br><b>第十一章 软件文件</b> ..... | <br>(269) |
| 11.1 目的和作用 .....           | (269)     |
| 11.2 软件生存周期与各种文件的编制 .....  | (270)     |
| 11.3 文件编制中考虑的因素 .....      | (271)     |
| 11.4 各种文件的内容要求 .....       | (274)     |
| 11.4.1 用户手册 .....          | (274)     |
| 11.4.2 操作手册 .....          | (274)     |
| 11.4.3 模块开发卷宗 .....        | (275)     |
| 11.4.4 开发进度月报 .....        | (276)     |
| 11.4.5 项目开发总结报告 .....      | (277)     |
| 11.5 文件编制实施规定的实例 .....     | (277)     |
| 11.6 文件编制的质量要求 .....       | (280)     |
| 11.7 文件的管理和维护 .....        | (281)     |
| 11.7.1 文件的形成 .....         | (281)     |
| 11.7.2 文件的分类与标识 .....      | (282)     |
| 11.7.3 文件控制 .....          | (282)     |
| 11.7.4 文件的修改管理 .....       | (283)     |
| 11.8 文件编制工具 .....          | (283)     |
| 11.9 小结 .....              | (283)     |
| 习题与思考 .....                | (284)     |
| <br><b>第十二章 软件维护</b> ..... | <br>(286) |
| 12.1 软件维护概述 .....          | (286)     |
| 12.1.1 软件维护工作的必要性 .....    | (286)     |
| 12.1.2 软件维护的内容 .....       | (287)     |

|                               |              |
|-------------------------------|--------------|
| 12.1.3 维护工作的过程 .....          | (288)        |
| 12.1.4 影响维护工作的因素 .....        | (289)        |
| 12.1.5 维护策略 .....             | (290)        |
| 12.2 软件的可维护性 .....            | (294)        |
| 12.2.1 可维护性的度量 .....          | (294)        |
| 12.2.2 提高软件的可维护性 .....        | (300)        |
| 12.3 软件维护的副作用 .....           | (305)        |
| 12.4 软件维护工具 .....             | (306)        |
| 12.5 软件维护的管理 .....            | (308)        |
| 12.5.1 软件维护的文件 .....          | (310)        |
| 12.6 软件逆向工程 .....             | (313)        |
| 12.6.1 反汇编、反编译 .....          | (313)        |
| 12.6.2 程序分析技术 .....           | (315)        |
| 12.6.3 功能分析技术 .....           | (317)        |
| 12.6.4 数据字典逆向自动生成 .....       | (318)        |
| 12.6.5 程序结构图逆向自动生成 .....      | (325)        |
| 12.7 软件重用技术 .....             | (331)        |
| 12.7.1 重用技术概述 .....           | (331)        |
| 12.7.2 可重用软件设计过程 .....        | (333)        |
| 12.8 小结 .....                 | (335)        |
| 习题与思考 .....                   | (336)        |
| <br>                          |              |
| <b>第十三章 软件工具和软件开发环境</b> ..... | <b>(340)</b> |
| 13.1 软件开发工具 .....             | (340)        |
| 13.2 软件开发环境 .....             | (342)        |
| 13.3 CASE 技术综述 .....          | (344)        |
| 13.3.1 CASE 的背景 .....         | (344)        |
| 13.3.2 CASE 工具在软件开发中的作用 ..... | (345)        |
| 13.3.3 CASE 环境框架 .....        | (346)        |
| 13.4 CASE 技术与方法学 .....        | (347)        |
| 13.5 CASE 的用户界面 .....         | (349)        |
| 13.6 CASE 工具的分类及应用 .....      | (350)        |
| 13.6.1 CASE 工具的分类 .....       | (350)        |
| 13.6.2 应用 CASE 工具的策略 .....    | (353)        |
| 13.6.3 CASE 工具的应用环境 .....     | (353)        |
| 13.7 CASE 技术的发展 .....         | (354)        |
| 13.7.1 第一代 CASE .....         | (354)        |
| 13.7.2 第二代 CASE .....         | (355)        |
| 13.7.3 CASE 与第四代语言 .....      | (357)        |

|                            |              |
|----------------------------|--------------|
| 13.7.4 CASE 与其他技术的结合 ..... | (358)        |
| 13.7.5 CASE 的应用与发展 .....   | (358)        |
| 13.8 分布式处理系统 .....         | (359)        |
| 13.8.1 分布式处理系统的概念 .....    | (360)        |
| 13.8.2 分布式处理系统的类型 .....    | (360)        |
| 13.8.3 分布式处理系统的效用 .....    | (362)        |
| 13.8.4 分布式数据库 .....        | (363)        |
| 13.9 小结 .....              | (365)        |
| 习题与思考 .....                | (366)        |
| <br>                       |              |
| <b>第十四章 软件的质量与评价</b> ..... | <b>(367)</b> |
| 14.1 软件质量度量模型 .....        | (367)        |
| 14.2 软件质量评价过程模型 .....      | (369)        |
| 14.3 软件质量度量元的选择 .....      | (370)        |
| 14.4 软件复杂性的度量 .....        | (371)        |
| 14.4.1 代码行数度量法 .....       | (371)        |
| 14.4.2 Halstead 度量法 .....  | (372)        |
| 14.4.3 McCabe 度量法 .....    | (374)        |
| 14.4.4 软件复杂性的综合度量 .....    | (375)        |
| 14.4.5 软件复杂性度量方法评价 .....   | (376)        |
| 14.5 软件可靠性评价 .....         | (377)        |
| 14.6 软件性能评价 .....          | (377)        |
| 14.7 软件评价 .....            | (378)        |
| 14.8 小结 .....              | (380)        |
| 习题与思考 .....                | (380)        |
| <br>                       |              |
| <b>第十五章 软件管理</b> .....     | <b>(385)</b> |
| 15.1 软件项目的特点与软件管理职能 .....  | (385)        |
| 15.2 软件开发组织 .....          | (386)        |
| 15.3 软件计划管理 .....          | (388)        |
| 15.3.1 进度安排的方法 .....       | (388)        |
| 15.3.2 进度的控制与描述 .....      | (388)        |
| 15.4 标准化管理 .....           | (389)        |
| 15.5 软件配置管理 .....          | (392)        |
| 15.5.1 软件配置 .....          | (392)        |
| 15.5.2 配置标识 .....          | (393)        |
| 15.5.3 里程碑与变更控制 .....      | (393)        |
| 15.5.4 配置状态登录 .....        | (394)        |
| 15.5.5 配置审查 .....          | (395)        |

---

|                             |           |
|-----------------------------|-----------|
| 15.6 软件的产权保护 .....          | (395)     |
| 15.6.1 软件知识产权的法律保护 .....    | (395)     |
| 15.6.2 计算机软件是著作权保护的客体 ..... | (397)     |
| 15.6.3 软件著作权人享有的专有权利 .....  | (398)     |
| 15.6.4 软件著作权的登记 .....       | (400)     |
| 15.7 软件产业 .....             | (401)     |
| 15.8 小结 .....               | (402)     |
| 习题与思考 .....                 | (403)     |
| <br>附录 A 一个出错处理程序的实例 .....  | <br>(406) |
| <br>参考文献 .....              | <br>(411) |

# 第一章 软件工程概述

随着计算机系统的迅速发展和应用范围的日益广泛, 计算机软件的规模越来越大, 其复杂程度也不断增加。进入 60 年代以来, 人们开始逐渐认识到了确实存在着“软件危机”这样一个事实。例如:

- (1) 软件生产不能满足日益增长的需要。
- (2) 软件开发成本和开发进度的估计往往不准确。实际成本有时高出预计成本好几倍, 预计完工的时间往往推迟几个月, 甚至几年。
- (3) 软件开发人员和用户之间信息交流不充分, 用户对完成的软件满意度很低。
- (4) 软件价格昂贵, 软件成本在整个计算机系统中所占的比例急剧上升, 软件已成为许多计算机系统中花钱最多的项目。
- (5) 软件质量难以保证, 软件质量保证技术还没有真正应用到软件开发的全过程。
- (6) 软件可维护性差, 程序中的错误很难改正, 或者当硬件环境发生变化时, 想要进行的适应性或完善性维护都是极其困难的。

导致这一系列问题的一个重要原因, 就是由于软件的研制和维护本身是工程性的任务, 但软件人员所采取的方式却未能工程化。

为了克服软件危机, 促使人们开始考虑采用工程化方法和工程途径来研制和维护软件。60 年代末至 70 年代初开始, 逐渐发展起一组总称为“软件工程”的技术。这些技术把软件作为一个工程产品来处理: 它需要计划、分析、设计、实现、测试以及维护。软件工程学主要研究软件结构、软件设计方法论、软件工具、软件工程标准和规范, 以及与软件工程有关的理论等。软件产业的逐步建立, 也带动了软件工程这门新的学科的产生和发展。同时, 软件生产的系列化、产品化、工程化和标准化形成了软件产业的标志。

## 1.1 计算机系统工程

软件工程和硬件工程都可以看成是一门更广义的学科——计算机系统工程的一部分。

用于计算机硬件的工程技术是由电子设计技术发展起来的, 而且, 在几十年的时间里已经达到了比较成熟的水平, 硬件设计技术已经很好地确立。虽然制造方法仍在不断地改进, 可靠性则已是一种可以期待的现实而不再是一种愿望了。

但是, 计算机软件工程却还处于某种困境之中。在以计算机为基础的系统, 软件已经取代硬件成为系统中设计起来最困难、最不容易成功(按时完成和不超过预计的成本), 而且是最不易管理的部分。另一方面, 随着以计算机为基础的系统在数量、复杂程度和应用范围上的不断增长, 对软件的需求仍然有增无减。

### 1.1.1 计算机系统的发展

软件发展的历史是与计算机系统发展紧密相关的。

在计算机系统发展的初期（50~60年代），硬件经历了不断的变化，而软件则被多数人作为一种事后工作来看待，对它而言，几乎没有什么系统的方法可以遵循。在此期间，多数系统采用批处理工作方式。在这些系统中，硬件通常仅用来执行一个单一的程序，而这个程序又是为一个特定的应用目的而编制的。

在早期，当通用硬件已经成为平常的事情时，软件则仍然是为每一种用途分别设计的，它们的通用性是有限的。作为产品的软件（即为了销售给一个或多个用户而研制的程序）还处在它的幼年时期。大多数软件是由使用该软件的人或机构研制的，使软件带有较强烈的个人色彩：软件设计是在某个人的头脑中完成的一种隐含的过程。而且，往往没有软件说明书。

计算机系统发展的第二个时期跨越了从60年代中期到70年代中期这十年。多道程序设计、多用户系统引入了人一机对话的新概念。人机对话技术为计算机的应用开辟了新的天地，并使硬件和软件提高到更为精湛的新水平。实时系统能够在几毫秒内而不是几分钟内收集、分析和变换来自多个信息源的数据，进而控制处理过程并产生输出。联机辅助存储设备的发展导致了数据库管理系统的出现。

第二个时期也是以产品化软件的使用和“软件车间”的出现为特征的。人们开发软件是为了广泛销售。

随着以计算机为基础的系统日益增多，计算机软件库开始膨胀。本单位自己开发的软件和从外部购买来的软件产品堆集了几十万条源程序的语句。所有这些程序，即所有这些源语句，当检测出故障时就必须加以维护；当使用者的要求改变时，必须加以修改；或者要使软件在新购置的硬件上运行时，也必须修改。软件维护的开销越来越大，更坏的是，许多软件所带有的个人色彩使得它们实际上是不可维护的。“软件危机”开始了。

计算机系统发展的第三个时期从70年代初期开始。分布式系统（多个计算机、各机器并行执行和相互通信）极大地增加了以计算机为基础的系统的复杂性。由于微处理机和有关部件的功能越来越强而价格越来越低，因此，在最常用的计算机应用领域中，具有“嵌入智能”的产品取代了较大的计算机。

此外，微处理机的出现也使得人们可能以极低的成本实现复杂的逻辑功能。

硬件的迅速发展已经超过了人们提供支持软件的能力。在第三个发展时期，软件危机日益严重。用于维护软件的费用占了数据处理预算的50%以上，而软件开发的生产率又跟不上新系统对软件需求的步伐。为了对付不断增长的软件危机，软件工程开始得到认真的对待。

如今，计算机系统的发展正在向第四个时期过渡。从技术性应用向消费性市场过渡需要软件设计产业化，这只能通过计算机系统工程来完成。

### 1.1.2 计算机系统工程

计算机系统工程的主要内容是对系统所要求的功能加以揭示、分析，并把它们分配给系统的各个部分。计算机系统工程的概貌如图1.1所示。

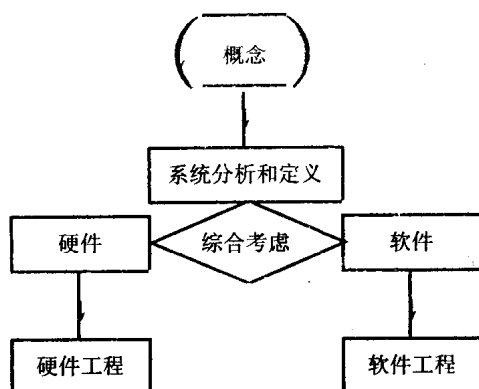


图 1.1 计算机系统工程

在大多数新系统创建时，对系统所要求的功能往往只有模糊的概念。系统分析和系统定义的目的在于揭示摆在面前的项目的范围。这就要对需要进行处理的信息、所要求的功能、所期望的性能以及设计的约束和检验的标准等进行系统的、详细的分析。

在范围确定之后，计算机系统工程师必须考虑多种能潜在地满足项目范围的、可供选择的配置。在综合考虑了各项因素之后，选择其中的一种配置，并将系统的功能分配给系统的各个部分。对一个以计算机为基础的系统来说，硬件和软件是最可能被选择的一些组成部分，就本书而言，我们主要讨论其中的软件工程部分。

## 1.2 软件和软件生存周期

首先，我们给软件下一个定义：

■ 软件是计算机系统上的程序和有关文件（《中国大百科全书》电子学与计算机Ⅱ）

也就是说，“软件”不仅仅是指程序，在软件研制过程中按一定规格产生的各种文件也是软件的不可缺少的组成部分。

一项计算机软件，从出现一个构思之日起，经过这项软件开发成功投入使用，在使用中不断增补修订，直到最后决定停止使用，并被另一项软件代替之时止，被认为是该软件的一个生存周期（或称生命周期，life cycle）。一个软件产品的生存周期可以划分成若干个互相区别而又有联系的阶段，每个阶段中的工作均以上一阶段工作的结果为依据，并为下一阶段的工作提供了前提。经验表明，失误造成的差错越是发生在生存周期前期，在系统交付使用时造成的影响和损失越大，要纠正它所花费的代价也越高。因而在前一阶段工作没有做好之前，决不要草率地进入下一阶段。软件生存周期阶段的划分，有助于软件研制管理人员借用传统工程的管理方法（重视工程性文件的编制，采用专业化分工方法，在不同阶段使用不同的人员等），从而有利于明显提高软件质量、降低成本、合理使用人才、进而提高软件开发的劳动生产率。