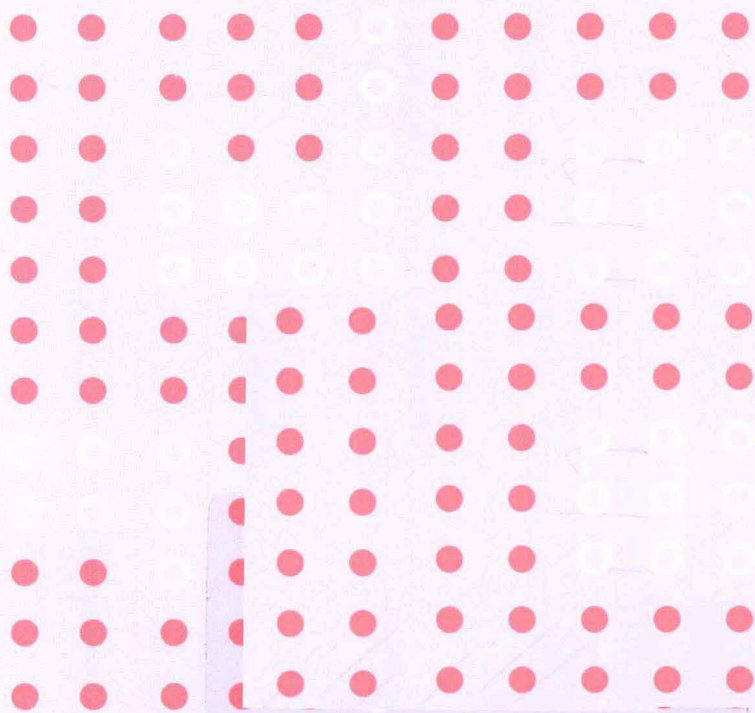


软件工程系列教材

田淑梅 廉龙颖 高辉 编著

软件工程

——理论与实践



软
件
工
程

<http://www.tup.com.cn>

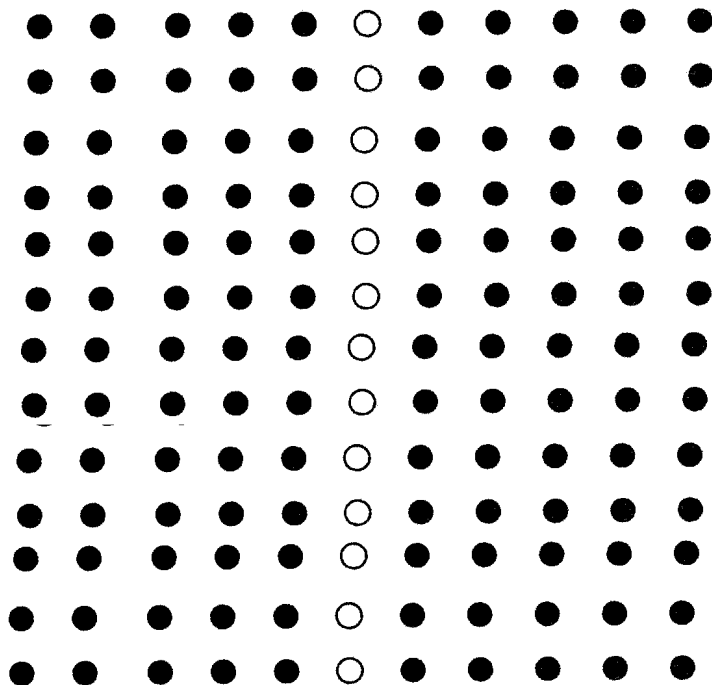
清华大学出版社

软件工程系列教材

软件工程

——理论与实践

田淑梅 廉龙颖 高辉 编著



清华大学出版社
北京

内 容 简 介

本书系统地介绍了软件工程的概 念、原理、方法与案例,全书共分 15 章,内容包括软件工程的基本概念及软件开发过程的各种模型、软件需求工程、面向数据流的需求分析、面向对象的方法及 UML 建模语言、面向对象的分析、软件设计工程、面向数据流的设计、面向对象的设计、软件实现、软件测试、软件维护、软件项目管理、软件复用技术、软件工程质量、软件工程标准化与软件文档等内容。书中各章涵盖了丰富的案例及习题,有助于学生理解和运用。

本书适合作为高等院校计算机专业或信息类相关专业本科生或研究生教材,也可作为软件开发人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件工程——理论与实践 / 田淑梅等编著. —北京:清华大学出版社,2011.9
(软件工程系列教材)

ISBN 978-7-302-25470-6

I. ①软… II. ①田… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 084263 号

责任编辑:焦虹 薛阳

责任校对:白蕾

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京四季青印刷厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185×260

印 张:27

字 数:641 千字

版 次:2011 年 9 月第 1 版

印 次:2011 年 9 月第 1 次印刷

印 数:1~3000

定 价:39.00 元

前言

PREFACE

自 1968年首次提出“软件工程”的概念至今，已经过去了四十多年，这期间软件工程得到了很大的发展。人们经历了多次软件危机后逐渐认识到软件工程在开发高质量软件产品中的重要性，并不断地探索软件工程的新方法、新技术和模型。随着计算机科学技术的飞速发展，软件工程已经成为这门学科的重要研究方向。

软件工程是高等学校计算机科学与技术学科各专业的一门重要的专业基础课，它的研究范围非常广泛，包括技术、方法、工具和管理等许多方面。严格遵循软件工程的方法，可以大大提高软件的开发效率和成功率。因此，本书在介绍软件工程的基本概念和基本理论的基础上，重点通过实例介绍软件开发的方法与技术，旨在通过本书的学习使读者能更好地运用软件工程方法开发出优质的软件。

本书文字通俗易懂、概念清晰、实例丰富、实用性强，并配有大量的习题，既可作为高等学校计算机专业课程的教材或参考书，也可以作为通信、电子信息等相关专业的软件工程课程教材，还可以供软件工程师、项目管理者 and 应用软件开发人员阅读参考。

本书的编者都是长期在高校从事软件工程教学的教师，具有丰富的教学经验和科研开发能力。本书共 15 章，其中第 1、4、5、8、11、14、15 章由田淑梅老师编写，田淑梅老师同时负责全书的统稿；第 2、6、9、12 章由廉龙颖老师编写；第 3、7、10、13 章由高辉老师编写。

由于软件工程发展迅速，加之作者的时间和水平有限，书中可能存在一些不足之处，希望读者提出宝贵的意见。

编者

2011 年 4 月

目 录

CONTENTS

| | |
|-------------------------|----|
| 第 1 章 软件工程概论 | 1 |
| 1.1 软件的概念 | 1 |
| 1.1.1 软件发展历程 | 1 |
| 1.1.2 软件的定义 | 2 |
| 1.1.3 软件的特点 | 3 |
| 1.1.4 软件分类 | 3 |
| 1.2 软件危机 | 5 |
| 1.2.1 软件危机的定义 | 5 |
| 1.2.2 产生软件危机的原因 | 5 |
| 1.2.3 软件危机的表现形式 | 5 |
| 1.2.4 解决软件危机的途径 | 6 |
| 1.3 软件工程的产生和发展 | 6 |
| 1.3.1 软件工程的定义及目标 | 6 |
| 1.3.2 软件工程的研究内容 | 8 |
| 1.3.3 软件工程的基本原理 | 9 |
| 1.4 软件过程和软件生存周期 | 10 |
| 1.4.1 软件过程 | 10 |
| 1.4.2 软件生存周期 | 11 |
| 1.5 软件过程模型 | 13 |
| 1.5.1 瀑布模型 | 13 |
| 1.5.2 快速原型模型 | 15 |
| 1.5.3 螺旋模型 | 16 |
| 1.5.4 增量模型 | 18 |
| 1.5.5 喷泉模型 | 19 |
| 1.5.6 基于构件的开发模型 | 20 |
| 1.5.7 统一过程(RUP)模型 | 21 |
| 1.5.8 形式化方法模型 | 25 |

| | | |
|------------|-------------------|-----------|
| 1.6 | 软件开发方法和软件开发工具 | 25 |
| 1.6.1 | 结构化方法 | 25 |
| 1.6.2 | 面向数据结构的软件开发方法 | 26 |
| 1.6.3 | 面向对象的方法 | 27 |
| 1.6.4 | 问题分析法 | 27 |
| 1.6.5 | 软件开发工具 | 27 |
| 1.7 | 传统的软件工程和面向对象的软件工程 | 28 |
| 1.7.1 | 传统的软件工程 | 28 |
| 1.7.2 | 面向对象的软件工程 | 29 |
| 1.8 | 本章小结 | 30 |
| 1.9 | 习题1 | 30 |
| 第2章 | 软件需求工程 | 33 |
| 2.1 | 软件需求工程概述 | 33 |
| 2.1.1 | 需求的概念 | 33 |
| 2.1.2 | 需求工程 | 34 |
| 2.2 | 需求获取 | 37 |
| 2.2.1 | 存在问题 | 37 |
| 2.2.2 | 综合需求 | 38 |
| 2.2.3 | 获取方法 | 39 |
| 2.2.4 | 提高获取的效率 | 41 |
| 2.2.5 | 案例：图书馆管理信息系统 | 44 |
| 2.3 | 需求分析建模 | 45 |
| 2.3.1 | 需求分析的主要工作 | 45 |
| 2.3.2 | 什么是需求分析模型 | 46 |
| 2.3.3 | 建模分类 | 46 |
| 2.3.4 | 分析建模的方法 | 47 |
| 2.4 | 需求规约 | 48 |
| 2.4.1 | 需求规格说明编写方法 | 48 |
| 2.4.2 | 需求规格说明书的主要内容 | 48 |
| 2.4.3 | 需求规格说明模板 | 48 |
| 2.4.4 | 需求文档的编写原则 | 52 |
| 2.4.5 | 案例：“软件工程”课程教学系统 | 52 |
| 2.5 | 需求评审 | 55 |
| 2.5.1 | 评审类型 | 55 |
| 2.5.2 | 评审标准 | 56 |
| 2.5.3 | 如何做好需求评审 | 57 |
| 2.6 | 需求管理 | 59 |

| | | |
|--------------|--------------------------------|-----------|
| 2.6.1 | 需求变更控制 | 59 |
| 2.6.2 | 需求跟踪 | 60 |
| 2.6.3 | 需求管理工具 | 62 |
| 2.7 | 本章小结 | 62 |
| 2.8 | 习题 2 | 63 |
| 第 3 章 | 结构化分析 | 65 |
| 3.1 | 概述 | 65 |
| 3.1.1 | 结构化分析思想 | 65 |
| 3.1.2 | 结构化分析过程 | 66 |
| 3.1.3 | 结构化模型的描述形式 | 67 |
| 3.2 | 数据流图 | 68 |
| 3.2.1 | 数据流图的基本成分 | 68 |
| 3.2.2 | 数据流图绘制方法 | 70 |
| 3.2.3 | 数据流图绘制实例 | 72 |
| 3.3 | 数据字典 | 77 |
| 3.3.1 | 数据字典的定义与用途 | 77 |
| 3.3.2 | 内容及格式 | 77 |
| 3.3.3 | 数据字典的实现 | 81 |
| 3.4 | 加工逻辑的描述工具 | 82 |
| 3.4.1 | 结构化语言 | 82 |
| 3.4.2 | 判定表 | 84 |
| 3.4.3 | 判定树 | 86 |
| 3.5 | 实体-关系图 | 86 |
| 3.5.1 | 数据对象 | 87 |
| 3.5.2 | 属性 | 87 |
| 3.5.3 | 关系 | 87 |
| 3.5.4 | 实体关系图的实现 | 88 |
| 3.6 | 本章小结 | 89 |
| 3.7 | 习题 3 | 89 |
| 第 4 章 | 面向对象的方法及 UML 建模语言 | 93 |
| 4.1 | 面向对象的方法学概述 | 93 |
| 4.1.1 | 面向对象方法的发展历程 | 93 |
| 4.1.2 | 面向对象方法学的优点 | 94 |
| 4.1.3 | 面向对象方法学的不足 | 96 |
| 4.2 | 面向对象的基本概念 | 97 |
| 4.3 | 面向对象的软件工程 | 100 |

| | | |
|------------|------------------------|------------|
| 4.4 | 面向对象建模 | 102 |
| 4.4.1 | 对象模型 | 104 |
| 4.4.2 | 动态模型 | 109 |
| 4.4.3 | 功能模型 | 110 |
| 4.4.4 | 3种模型之间的关系 | 113 |
| 4.5 | 典型的面向对象开发方法 | 113 |
| 4.5.1 | Booch方法 | 113 |
| 4.5.2 | Coad方法 | 114 |
| 4.5.3 | OMT方法 | 116 |
| 4.5.4 | OOSE方法 | 117 |
| 4.6 | UML统一建模语言 | 118 |
| 4.6.1 | UML语言的发展过程 | 119 |
| 4.6.2 | UML的定义及主要内容 | 119 |
| 4.6.3 | UML的主要特点 | 121 |
| 4.6.4 | UML的应用 | 122 |
| 4.6.5 | UML的本质和目标 | 122 |
| 4.7 | UML语言概述 | 123 |
| 4.7.1 | 视图 | 123 |
| 4.7.2 | UML的模型元素和表示方法 | 124 |
| 4.7.3 | 图 | 124 |
| 4.7.4 | UML建模过程 | 140 |
| 4.8 | 本章小结 | 141 |
| 4.9 | 习题4 | 141 |
| 第5章 | 面向对象的分析 | 145 |
| 5.1 | 面向对象的分析过程 | 145 |
| 5.1.1 | 面向对象分析的3个模型与5个层次 | 146 |
| 5.1.2 | 需求陈述 | 147 |
| 5.2 | 建立对象模型 | 148 |
| 5.2.1 | 确定类和对象 | 148 |
| 5.2.2 | 确定关联 | 150 |
| 5.2.3 | 确定属性 | 153 |
| 5.2.4 | 划分主题 | 154 |
| 5.2.5 | 识别结构 | 156 |
| 5.2.6 | 优化对象模型 | 156 |
| 5.3 | 建立动态模型 | 158 |
| 5.3.1 | 编写脚本 | 159 |
| 5.3.2 | 设想用户界面 | 160 |

| | | |
|--------------|----------------|------------|
| 5.3.3 | 画事件跟踪图 | 160 |
| 5.3.4 | 画状态图 | 161 |
| 5.3.5 | 优化动态模型 | 163 |
| 5.4 | 建立功能模型 | 165 |
| 5.5 | 面向对象分析实例 | 167 |
| 5.6 | 本章小结 | 172 |
| 5.7 | 习题 5 | 172 |
| 第 6 章 | 软件设计工程 | 175 |
| 6.1 | 软件设计工程概述 | 175 |
| 6.1.1 | 软件设计模型 | 176 |
| 6.1.2 | 设计目标和原则 | 176 |
| 6.1.3 | 软件设计基本方法 | 178 |
| 6.1.4 | 设计文档 | 183 |
| 6.2 | 数据设计 | 189 |
| 6.2.1 | 数据三模式的概念 | 189 |
| 6.2.2 | 数据库设计的原则 | 191 |
| 6.2.3 | 数据仓库 | 193 |
| 6.3 | 体系结构设计 | 194 |
| 6.3.1 | 体系结构设计准则 | 195 |
| 6.3.2 | 体系结构设计可采用的图形工具 | 196 |
| 6.4 | 人机界面设计 | 199 |
| 6.4.1 | 人机界面风格 | 199 |
| 6.4.2 | 人机界面设计原则 | 199 |
| 6.4.3 | 人机界面设计过程 | 200 |
| 6.5 | 详细设计 | 202 |
| 6.5.1 | 详细设计的任务 | 202 |
| 6.5.2 | 详细设计的常用工具 | 203 |
| 6.6 | 本章小结 | 207 |
| 6.7 | 习题 6 | 207 |
| 第 7 章 | 结构化设计 | 211 |
| 7.1 | 概述 | 211 |
| 7.1.1 | 软件设计的任务 | 211 |
| 7.1.2 | 结构化设计与结构化分析的关系 | 212 |
| 7.2 | 面向数据流的设计方法 | 213 |
| 7.2.1 | 设计过程 | 213 |
| 7.2.2 | 数据流分类 | 214 |

| | | |
|--------------|-------------------|------------|
| 7.2.3 | 变换分析 | 216 |
| 7.2.4 | 事务分析 | 219 |
| 7.2.5 | 综合分析 | 220 |
| 7.3 | 面向数据结构的设计方法 | 221 |
| 7.3.1 | Jackson 图 | 222 |
| 7.3.2 | Jackson 方法设计步骤 | 224 |
| 7.3.3 | 实例 | 224 |
| 7.4 | 本章小结 | 228 |
| 7.5 | 习题 7 | 228 |
| 第 8 章 | 面向对象的设计与实现 | 231 |
| 8.1 | 面向对象设计准则 | 231 |
| 8.2 | 启发式规则 | 233 |
| 8.3 | 系统设计 | 234 |
| 8.3.1 | 问题域子系统设计 | 236 |
| 8.3.2 | 人机交互子系统设计 | 237 |
| 8.3.3 | 任务管理子系统设计 | 241 |
| 8.3.4 | 数据管理子系统设计 | 243 |
| 8.4 | 服务与关联设计 | 247 |
| 8.4.1 | 设计服务 | 247 |
| 8.4.2 | 设计关联 | 248 |
| 8.5 | 面向对象设计优化 | 250 |
| 8.6 | 面向对象设计实例 | 253 |
| 8.7 | 面向对象系统的实现 | 259 |
| 8.7.1 | 选择程序设计语言 | 259 |
| 8.7.2 | 面向对象程序设计风格 | 262 |
| 8.8 | 本章小结 | 263 |
| 8.9 | 习题 8 | 264 |
| 第 9 章 | 软件实现 | 267 |
| 9.1 | 程序设计语言 | 268 |
| 9.1.1 | 程序设计语言的特性 | 268 |
| 9.1.2 | 程序设计语言的基本机制 | 269 |
| 9.1.3 | 程序设计语言的类型 | 270 |
| 9.1.4 | 程序设计语言的选择 | 271 |
| 9.2 | 程序设计风格 | 272 |
| 9.3 | 本章小结 | 274 |

| | | |
|---------------|-------------------|------------|
| 9.4 | 习题 9 | 274 |
| 第 10 章 | 软件测试 | 277 |
| 10.1 | 软件测试概述 | 277 |
| 10.1.1 | 软件测试的目的 | 277 |
| 10.1.2 | 软件测试的原则 | 278 |
| 10.1.3 | 软件测试分类 | 279 |
| 10.2 | 白盒测试 | 280 |
| 10.2.1 | 逻辑覆盖法 | 280 |
| 10.2.2 | 循环覆盖法 | 284 |
| 10.2.3 | 基本路径法 | 286 |
| 10.3 | 黑盒测试 | 288 |
| 10.3.1 | 等价类划分法 | 289 |
| 10.3.2 | 边界值分析法 | 291 |
| 10.3.3 | 因果图法 | 292 |
| 10.4 | 测试过程 | 295 |
| 10.4.1 | 单元测试 | 296 |
| 10.4.2 | 集成测试 | 297 |
| 10.4.3 | 确认测试 | 298 |
| 10.4.4 | 系统测试 | 300 |
| 10.4.5 | 验收测试 | 301 |
| 10.5 | 面向对象测试 | 302 |
| 10.5.1 | 面向对象测试策略 | 302 |
| 10.5.2 | 面向对象测试用例设计 | 310 |
| 10.6 | 自动化测试 | 312 |
| 10.6.1 | 测试自动化的基础 | 312 |
| 10.6.2 | 测试工具选型原则 | 313 |
| 10.6.3 | 自动化测试工具 | 314 |
| 10.7 | 软件调试 | 317 |
| 10.7.1 | 调试过程 | 318 |
| 10.7.2 | 调试技术 | 319 |
| 10.8 | 本章小结 | 320 |
| 10.9 | 习题 10 | 320 |
| 第 11 章 | 软件维护 | 323 |
| 11.1 | 软件维护概述 | 323 |
| 11.1.1 | 软件维护的基本概念 | 323 |
| 11.1.2 | 软件维护的种类 | 323 |

| | | |
|---------------|---------------|------------|
| 11.1.3 | 影响软件维护的工作量的因素 | 324 |
| 11.1.4 | 软件维护策略 | 325 |
| 11.1.5 | 维护的副作用 | 326 |
| 11.2 | 软件维护过程 | 327 |
| 11.2.1 | 建立维护机构 | 327 |
| 11.2.2 | 编写软件维护申请报告 | 328 |
| 11.2.3 | 软件维护工作流程 | 328 |
| 11.2.4 | 软件维护记录及评价 | 329 |
| 11.3 | 软件可维护性 | 330 |
| 11.3.1 | 可维护性定义 | 330 |
| 11.3.2 | 可维护性度量 | 331 |
| 11.3.3 | 提高软件可维护性方法 | 333 |
| 11.4 | 逆向工程和再工程 | 337 |
| 11.4.1 | 软件再工程过程 | 337 |
| 11.4.2 | 软件再工程分析 | 339 |
| 11.5 | 本章小结 | 340 |
| 11.6 | 习题 11 | 341 |
| 第 12 章 | 软件项目管理 | 343 |
| 12.1 | 软件项目管理概述 | 343 |
| 12.1.1 | 软件项目管理的定义 | 343 |
| 12.1.2 | 软件项目管理的过程 | 344 |
| 12.1.3 | 软件项目管理的内容 | 345 |
| 12.2 | 软件项目成本管理 | 345 |
| 12.2.1 | 成本管理 | 345 |
| 12.2.2 | 软件项目成本估算方法 | 346 |
| 12.2.3 | 软件项目成本估算模型 | 347 |
| 12.3 | 软件项目进度管理 | 348 |
| 12.3.1 | 软件项目进度管理内容 | 348 |
| 12.3.2 | 制订项目计划 | 348 |
| 12.3.3 | 项目进度监控 | 349 |
| 12.4 | 软件项目配置管理 | 350 |
| 12.4.1 | 软件配置管理过程 | 350 |
| 12.4.2 | 软件配置管理职责 | 351 |
| 12.5 | 软件项目风险管理 | 352 |
| 12.5.1 | 软件项目中的风险种类 | 352 |
| 12.5.2 | 软件项目风险管理 | 354 |
| 12.5.3 | 风险识别 | 354 |

| | | |
|---------------|---------------|------------|
| 12.5.4 | 风险评估 | 355 |
| 12.5.5 | 风险跟踪 | 356 |
| 12.5.6 | 风险处理 | 357 |
| 12.6 | 软件项目质量管理 | 358 |
| 12.7 | 软件项目人员管理 | 359 |
| 12.7.1 | 人员管理工作内容 | 359 |
| 12.7.2 | 项目参与者 | 360 |
| 12.7.3 | 团队建设 | 360 |
| 12.8 | 本章小结 | 361 |
| 12.9 | 习题 12 | 361 |
| 第 13 章 | 软件复用技术 | 365 |
| 13.1 | 概述 | 365 |
| 13.1.1 | 复用的概念 | 365 |
| 13.1.2 | 复用的意义 | 367 |
| 13.1.3 | 如何实现复用 | 368 |
| 13.1.4 | 软件复用技术 | 368 |
| 13.1.5 | 可复用的软件成分 | 369 |
| 13.2 | 软件复用过程 | 370 |
| 13.2.1 | 构件组装模型 | 370 |
| 13.2.2 | 复用模型 | 371 |
| 13.2.3 | 软件复用过程模型 | 372 |
| 13.3 | 软件构件技术 | 373 |
| 13.3.1 | 软件构件技术产生背景 | 373 |
| 13.3.2 | 软件构件基本概念 | 373 |
| 13.3.3 | 软件构件的基本特征 | 374 |
| 13.4 | 领域工程 | 375 |
| 13.4.1 | 领域划分 | 375 |
| 13.4.2 | 领域分析过程 | 375 |
| 13.4.3 | 领域特征 | 376 |
| 13.4.4 | 结构化建模与结构点 | 377 |
| 13.5 | 开发可复用的软件构件 | 377 |
| 13.5.1 | 为了复用的分析与设计 | 377 |
| 13.5.2 | 可复用构件的设计准则 | 378 |
| 13.5.3 | 基于构件的开发 | 378 |
| 13.6 | 分类和检索软件构件 | 380 |
| 13.6.1 | 描述可复用的构件 | 380 |
| 13.6.2 | 复用环境 | 382 |

| | | |
|---------------|---------------------|------------|
| 13.7 | 软件复用的效益 | 382 |
| 13.8 | 本章小结 | 383 |
| 13.9 | 习题 13 | 383 |
| 第 14 章 | 软件工程质量 | 385 |
| 14.1 | 软件质量特性 | 385 |
| 14.1.1 | 软件质量的定义 | 385 |
| 14.1.2 | 软件质量的特性 | 386 |
| 14.2 | 软件质量的度量 | 386 |
| 14.2.1 | 软件度量 | 387 |
| 14.2.2 | 软件度量的作用 | 387 |
| 14.2.3 | 软件度量的分类 | 388 |
| 14.2.4 | 软件度量过程 | 388 |
| 14.3 | 软件质量保证 | 389 |
| 14.3.1 | 质量控制与质量保证 | 390 |
| 14.3.2 | 软件质量保证的内容 | 391 |
| 14.3.3 | 软件质量保证策略 | 392 |
| 14.4 | 技术评审 | 392 |
| 14.4.1 | 准备评审 | 393 |
| 14.4.2 | 举行评审会议 | 393 |
| 14.4.3 | 跟踪与审核 | 394 |
| 14.5 | 软件质量管理体系 | 394 |
| 14.5.1 | ISO 9000 系列标准 | 394 |
| 14.5.2 | 软件质量认证 | 395 |
| 14.6 | CMM 模型 | 395 |
| 14.6.1 | CMM 的结构 | 395 |
| 14.6.2 | 软件过程能力成熟度等级 | 396 |
| 14.6.3 | 关键过程域 | 397 |
| 14.6.4 | 关键实践 | 398 |
| 14.7 | 本章小结 | 399 |
| 14.8 | 习题 14 | 399 |
| 第 15 章 | 软件工程标准化与软件文档 | 401 |
| 15.1 | 软件工程标准化 | 401 |
| 15.1.1 | 软件工程标准化的概念 | 402 |
| 15.1.2 | 软件工程标准化的意义 | 402 |
| 15.1.3 | 软件工程化的类型 | 403 |
| 15.1.4 | 软件工程标准的制定与推行 | 404 |

| | | |
|-------------------|------------------|------------|
| 15.2 | 软件组织内的标准化工作····· | 405 |
| 15.3 | 软件文档与编写要求····· | 406 |
| 15.3.1 | 软件文档的作用和分类····· | 406 |
| 15.3.2 | 软件文档的编写方法····· | 409 |
| 15.3.3 | 软件基本文档的内容要求····· | 409 |
| 15.3.4 | 对档编制的质量要求····· | 413 |
| 15.3.5 | 软件文档的管理和维护····· | 413 |
| 15.4 | 本章小结····· | 414 |
| 15.5 | 习题 15 ····· | 414 |
| 参考文献 ····· | | 416 |

本章主要讲述软件的发展历程、定义及特征,软件危机的表现及解决途径,软件工程的定义及三要素,软件工程的基本原理及目标,软件生存周期的概念和内容,8种软件开发模型,软件开发方法。本章难点是软件开发模型的使用。

本章学习目标:

- 掌握软件的定义,软件工程的定义及三要素。
- 掌握软件生存周期的概念和内容,8种软件开发模型。
- 了解软件危机的表现及解决途径,软件工程的基本原理及目标,软件开发方法。

1.1 软件的概念

计算机软件的发展与进步,与计算机硬件的发展和计算机的普及是分不开的。同其他事物的发展规律一样,它也经历了从产生、发展到成熟的过程。在这个过程中也经历了软件的危机,为吸取历史经验教训,应该认真研究产生软件危机的原因,探讨消除软件危机的途径。

1.1.1 软件发展历程

17世纪60年代,Ada(Augusta Ada Lovelace)成为历史上第一个写软件的人,她第一次为Babbage(Charles Babbage)的分析机编制程序,实现了计算伯努利函数、级数相乘函数等功能。也因此,人们公认她是世界上第一位软件工程师。20世纪40年代末,第一台电子计算机ENIAC(Electronic Numerical Integrator And Calculator)在美国问世,软件也随之诞生,从此,很多软件工程师陆续出现。到了20世纪60年代,美国出现了专门编写软件的专业,伴随着信息产业的迅速发展,软件对人类社会也起到了越来越突出的作用,人们对软件的认识也更为深刻。

软件的发展主要经历了以下3个发展阶段。

1. 第一阶段(20世纪50年代初期至20世纪60年代中期)

这个阶段也被称为程序设计阶段。软件生产以个体化为主;计算机的存储容量小,运算速度比较慢;编写程序的工具只有低级语言;软件规模不大,几乎没有系统化的标准可遵循,对软件开发也没有良好的管理方法。大多数的软件由软件使用者自己开发和编写,然后个人应用。当时并没有“软件”一词,人们对与程序有关的文档的重要性认识不足,开发工作主要围绕硬件进行。工程规模很小,使用工具单一,开发者之间更没有明确分工。

2. 第二阶段(20世纪60年代中期至20世纪70年代末期)

这个阶段也可称为程序系统阶段。随着计算机硬件的飞速发展,出现了大容量的存储器;出现了高级程序设计语言,计算机的应用领域也不断扩大;多道程序设计、用户系统引入了人机交互的新概念;出现了实时系统和第一代数据库管理系统。这个阶段的另一个特点就是软件产品的使用和软件作坊的出现。软件的应用范围更加广泛,一个程序可以有多个用户使用。

在软件使用中,由于缺乏有效的工程化方法的指导,很多软件不能按计划完成,纠正源代码错误、变更程序功能等软件维护花费了人们更多的精力和资源,到了20世纪60年代中期,出现了所谓的软件危机。

3. 第三阶段(20世纪70年代末至今)

这个阶段,计算机硬件向巨型机和微型机两个方面发展;出现了计算机网络,特别是Internet得到了飞速发展;软件方面出现了软件工程;随着分布式系统、高带宽数字通信系统、实时数据访问控制系统等应用技术的迅速发展,人们对计算机软件的需求变得更高,同时也使得软件开发的效率和质量成为人们关注的焦点。因此,以软件产品化、系列化、工程化、标准化为特征的软件产业迅猛发展,推动了软件工程学的进步。

1.1.2 软件的定义

软件是计算机系统中与硬件(hardware)相互依存的另一部分,与硬件合为一体完成系统功能。

软件定义包括如下几点。

- (1) 功能和性能的指令集(即程序)。
- (2) 程序能正常操纵信息的数据结构(即相关数据)。
- (3) 与程序开发维护和使用有关的各种图文数据(即说明文档)。

随着计算机应用的日益普及,软件变得越来越复杂,规模变得越来越大,为保证软件开发与维护工作顺利有序地进行,人与人、人与计算机间的相互沟通也越来越重要,因此,说明文档(即各种程序文本、报告、说明、手册等)在软件开发与维护中体现的价值超出了软件产品本身。特别是在软件日益成为产品的今天,文档的作用更加重要了。