



普通高等教育 电气信息类 应用型规划教材

软件工程教程

(第四版)

周 苏 张丽娜 王 文 编著



科学出版社



免费提供电子教案

普通高等教育电气信息类应用型规划教材

软件工程学教程

(第四版)

周 苏 张丽娜 王 文 编著

科学出版社

北 京

内 容 简 介

软件工程学是一门理论性和实践性都很强的学科,它采用工程化的概念、理论、技术和方法来指导开发与维护计算机软件。本书系统、全面地介绍了软件工程技术,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,具有较强的系统性和可读性;本书较为详细地介绍了软件生存周期、传统软件工程、软件工程过程、面向对象软件工程和软件体系结构等的思想和实现方法,并力图反映软件工程领域的最新发展。按照一系列软件工程国家标准来表达和描述软件工程的学问,使软件工程技术具有很强的可操作性,是本书的主要特色之一。

本书可作为高等院校“软件工程”课程的教材和教学参考书,也可供有一定实践经验的软件开发人员、管理人员参考,或作为继续教育的教材,还可作为各个级别的计算机软件专业技术资格和水平考试中相关内容的学习辅导用书。

图书在版编目(CIP)数据

软件工程学教程 / 周苏, 张丽娜, 王文编著. —4 版. —北京: 科学出版社, 2011

ISBN 978-7-03-012897-3

I. ①软… II. ①周…②张…③王… 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(2004)第 008747 号

责任编辑: 陈晓萍 / 责任校对: 柏连海

责任印制: 吕春珉 / 封面设计: 耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

*

2011 年 8 月第 四 版 开本: 787×1092 1/16

2011 年 8 月第七次印刷 印张: 22 1/4

印数: 18 001—20 000 字数: 522 000

定价: 38.00 元

(如有印装质量问题, 我社负责调换<新蕾>)

销售部电话 010-62142126 编辑部电话 010-62138978-8003

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

软件工程是用来指导计算机软件开发和维护的工程性学科,它主要研究软件结构、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。采用工程化的概念、原理、技术和方法来开发与维护软件,把经过时间考验而证明正确的管理技术和当前能够得到的最好的开发方法结合起来,这就是软件工程。

1992年,结合当时在杭州大学的教学实践,我们在天津科学技术出版社出版了《软件工程及其应用》一书。以此为基础,2002年,结合多年在应用型高等院校的软件工程课程教学实践,我们在科学出版社出版了《软件工程学教程》,亦即本书的第一版。按照软件工程国家标准来表达和描述软件工程的知识,使软件工程技术具有很强的可操作性,是本书的主要特色之一,出版后受到了广大师生的广泛欢迎。

基于软件工程国家标准的发展和软件生存周期过程思想的提出,为进一步按照过程来管理和实施软件工程实践,发展和完善软件工程方法,2004年我们出版了本书的第二版。

2008年,结合浙江大学城市学院“软件工程”精品课程建设和浙江省高等院校重点教材建设项目,我们在浙江科学技术出版社出版了《软件工程基础》一书,是以为本书的第三版。

伴随着软件工程技术和以软件工程为基础的课程体系建设的长足发展,我们的“软件工程”顺利成为杭州市精品课程,相应的教学改革成果荣获学院“教学成果奖”。以这一系列教学研究成果为基础,我们重新编写了本书的第四版。

本书是具有较强的实践性的高等院校“软件工程”课程教材。本书针对计算机和其他IT专业学生的发展需求,系统、全面地介绍了软件工程的概念、原理、方法及其应用,详细介绍了软件生存周期、传统软件工程、面向对象软件工程、软件过程工程和软件体系结构设计的思想与实现方法,力图反映软件工程领域的最新发展,具有较强的系统性和可读性。

本书的主要特色是:理论联系实际,把软件工程的概念、理论和技术知识融入实践中,使学生保持浓厚的学习热情,加深对软件工程知识的认识、理解和掌握;按照一系列软件工程国家标准来表达和描述软件工程的知识,使软件工程技术具有很强的可操作性。

本书选编了大量的思考与习题,可作为高等院校“软件工程”课程的主教材,也可供有一定实践经验的软件开发人员、管理人员参考或作为继续教育的教材。

朱勇、魏金岭、翁正秋、周元哲、陈晓辉等参加了本书的部分编写工作。本书的编撰得到了浙江大学城市学院、温州大学城市学院、浙江商业职业技术学院、西安邮电学院等院校的大力支持，在此一并表示感谢。欢迎教师索取为本书教学配套的相关资料并交流：

E-mail: zhousu@qq.com

QQ: 81505050

个人博客: <http://blog.sina.com.cn/zhousu58>

目 录

前言

第1章 软件工程概述

1.1	计算机系统与软件	1
1.2	软件生存周期和软件生存周期过程	2
1.3	软件生存周期模型	4
1.3.1	瀑布模型	4
1.3.2	渐增模型	5
1.3.3	演化模型	6
1.3.4	螺旋模型	7
1.3.5	喷泉模型	8
1.3.6	智能模型	8
1.4	软件工程定义	9
1.4.1	软件工程的内容	9
1.4.2	软件工程的基本目标和原则	10
1.4.3	软件工程与一般工程的差异	11
1.5	软件工具与开发环境	12
1.5.1	软件工具	12
1.5.2	软件开发环境	13
1.5.3	软件支持环境	14
1.6	软件工程发展的4个阶段	14
1.7	软件工程师的责任	15
1.7.1	IEEE/ACM《计算学科教学计划》的相关要求	15
1.7.2	软件工程职业道德规范和职业实践要求	16
1.8	小结	16
	思考与习题	17
第2章	系统定义与软件计划	19
2.1	系统定义	19
2.1.1	系统分析	20
2.1.2	可行性研究	25
2.1.3	成本-收益分析	25

2.1.4 功能分配	27
2.1.5 系统规格说明	27
2.1.6 系统定义复审	28
2.2 软件计划	29
2.2.1 软件的范围	29
2.2.2 资源	30
2.2.3 软件成本估算	32
2.3 进度安排	32
2.4 计划文件与复审	35
2.5 小结	37
思考与习题	37
第3章 软件需求分析	38
3.1 需求分析阶段的任务	38
3.2 结构化分析方法	39
3.3 数据流程图	41
3.4 数据字典	45
3.4.1 数据流条目	45
3.4.2 文件条目	46
3.4.3 数据项条目	46
3.4.4 加工条目	46
3.5 加工的分析与表达	47
3.5.1 结构化语言	48
3.5.2 判定表	49
3.5.3 判定树	51
3.5.4 3种表达工具的比较	51
3.6 需求分析文件与复审	52
3.6.1 GB/T8567—2006规定的文件	52
3.6.2 计算机软件需求说明编制指南	54
3.6.3 需求分析的复审	54
3.7 小结	55
思考与习题	55
第4章 软件概要设计	57
4.1 模块的划分	57
4.1.1 系统性能的衡量标准	57
4.1.2 软件结构	58
4.1.3 模块划分的基本原则	59
4.1.4 内聚度	59

4.1.5	耦合度	61
4.1.6	高内聚和低耦合	61
4.1.7	模块划分的方法	62
4.2	结构化设计方法	63
4.2.1	变换与事务型数据流分析	63
4.2.2	模块化设计	65
4.2.3	模块结构图	66
4.2.4	从数据流程图导出模块结构图	68
4.3	Parnas 方法	69
4.3.1	信息隐蔽原则	69
4.3.2	加强系统各成分间的检查	70
4.4	Jackson 方法	70
4.5	程序的逻辑构造方法	71
4.6	概要设计文件与复审	72
4.6.1	概要设计说明书	72
4.6.2	概要设计的复审	73
4.7	小结	73
	思考与习题	73
第 5 章	软件详细设计	77
5.1	概述	77
5.2	结构化构造	78
5.3	图形设计工具	79
5.3.1	程序流程图	79
5.3.2	方块图	81
5.3.3	HIPO 图	82
5.3.4	PAD 图	84
5.4	伪码与程序设计语言	85
5.5	各种详细设计工具的比较	88
5.6	详细设计文件与复审	89
5.6.1	详细设计说明书	89
5.6.2	详细设计的复审	90
5.7	小结	90
	思考与习题	90
第 6 章	软件界面设计	92
6.1	代码设计	92
6.1.1	代码基础	92
6.1.2	代码的种类	93
6.1.3	代码设计步骤	95
6.1.4	代码设计文件	96

6.2	输出设计	97
6.2.1	输出设计的研究范围	97
6.2.2	输出设计的规则	98
6.2.3	输出设计方法	99
6.3	输入设计	102
6.3.1	输入设计的研究范围	102
6.3.2	输入设计的规则	102
6.3.3	输入设计方法	103
6.4	用户界面设计	106
6.4.1	用户界面开发的基本原则	107
6.4.2	出错控制与处理	110
6.5	软件安全性设计	111
6.5.1	系统安全控制的目的	111
6.5.2	软件安全控制的基本方法	111
6.5.3	软件的安全控制设计	114
6.6	小结	116
	思考与习题	117
第7章	软件编码	118
7.1	结构化程序设计方法	118
7.2	程序设计风格	119
7.2.1	源程序	119
7.2.2	数据说明	120
7.2.3	语句结构	121
7.3	源代码文件	121
7.3.1	综合文件	121
7.3.2	程序组织文件	122
7.3.3	指令级注释	122
7.4	程序设计技术	123
7.4.1	冗余程序设计	123
7.4.2	防错性程序设计	123
7.4.3	程序设计的质量	124
7.4.4	编译程序和解释程序	124
7.5	编程语言的特点	125
7.5.1	过程性语言	125
7.5.2	说明性语言	125
7.5.3	脚本语言	126
7.5.4	低级语言	126
7.5.5	高级语言	126

7.5.6 面向对象语言	126
7.5.7 事件驱动语言	127
7.5.8 构件(组件)	127
7.6 选择编程语言	127
7.7 编码文件与复审	129
7.8 小结	129
思考与习题	130
第8章 软件测试	133
8.1 测试的基本概念	133
8.2 测试方法	135
8.2.1 静态分析技术	135
8.2.2 动态测试技术	136
8.3 单元测试	137
8.4 组装测试	139
8.5 确认测试	141
8.6 测试用例设计	143
8.6.1 白盒法	143
8.6.2 黑盒法	145
8.7 测试工具与测试自动化	147
8.7.1 基于 GUI 的自动化测试	147
8.7.2 自动化测试工具的特征	148
8.7.3 自动化测试工具的分类	149
8.7.4 α 、 β 测试	150
8.8 测试文件与复审	151
8.8.1 测试文件综述	151
8.8.2 GB/T8567—2006 规定的文件	152
8.8.3 GB/T9386—1988 计算机软件测试文件编制规范	154
8.8.4 GB/T15532—2008 计算机软件测试规范	156
8.8.5 测试文件编制指南	157
8.8.6 测试的复审	158
8.9 排错技术	160
8.10 系统转换	161
8.11 小结	162
思考与习题	162
第9章 面向对象分析与设计	168
9.1 面向对象方法	168
9.1.1 面向对象方法的特点	169
9.1.2 面向对象软件工程	170

9.2 面向对象的概念	171
9.2.1 对象	171
9.2.2 类	172
9.2.3 消息传递	172
9.2.4 多态性	172
9.3 面向对象软件的开发过程	172
9.3.1 类生存期	173
9.3.2 面向对象的开发	175
9.4 面向对象分析 OOA	175
9.4.1 OOA 的基本内容	176
9.4.2 常用的 OOA 方法	176
9.4.3 论域分析	178
9.5 OOA 的 Yourdon 方法	181
9.5.1 标识类和对象	181
9.5.2 细化类和对象	183
9.5.3 标识属性	183
9.5.4 标识服务	185
9.5.5 标识结构	185
9.5.6 标识主题	186
9.6 面向对象设计 OOD	187
9.6.1 高层设计	187
9.6.2 类设计的目标和方针	188
9.6.3 通过复用设计类	190
9.6.4 类设计方法	190
9.7 OOD 的 Yourdon 方法	192
9.7.1 问题论域部分的设计	192
9.7.2 用户界面部分的设计	193
9.7.3 任务管理部分的设计	195
9.7.4 数据管理部分的设计	196
9.8 Booch 的方法	197
9.9 小结	197
思考与习题	198
第10章 面向对象的实现	199
10.1 面向对象编程	199
10.1.1 程序设计语言的影响	200
10.1.2 面向对象编程语言	200
10.2 面向对象测试	201
10.2.1 面向对象测试的特点	201
10.2.2 面向对象的测试步骤	202

10.2.3	测试 OOA 和 OOD 模型	205
10.2.4	面向对象的测试策略	205
10.3	OO 软件的测试用例设计	206
10.3.1	传统测试用例设计方法的可用性	206
10.3.2	基于故障的测试	206
10.3.3	基于场景的测试设计	207
10.3.4	测试表层结构和深层结构	207
10.4	统一建模语言 UML	207
10.5	小结	208
	思考与习题	209
第 11 章	软件文件	210
11.1	目的和作用	210
11.2	软件生存周期与各种文件的编制	211
11.3	文件编制中考虑的因素	212
11.3.1	文件的读者	212
11.3.2	文件内容的重复性	213
11.3.3	文件内容的灵活性	214
11.4	各种文件的内容要求	215
11.4.1	用户手册	215
11.4.2	操作手册	215
11.4.3	模块开发卷宗	216
11.4.4	开发进度月报	217
11.4.5	项目开发总结报告	217
11.5	文件编制实施规定的实例	219
11.6	文件编制的质量要求	221
11.7	文件的管理和维护	222
11.7.1	文件的形成	222
11.7.2	文件的分类与标识	222
11.7.3	文件控制	223
11.7.4	文件的修改管理	223
11.7.5	软件文档管理指南	224
11.8	文件编制工具	224
11.9	小结	225
	思考与习题	225
第 12 章	软件维护	227
12.1	概述	227
12.1.1	软件维护工作的必要性	227
12.1.2	软件维护的内容	228

12.1.3	维护工作的过程	230
12.1.4	影响维护工作的因素	230
12.1.5	维护策略	231
12.2	软件的可维护性	236
12.2.1	可维护性的度量	236
12.2.2	提高软件的可维护性	241
12.3	软件维护工具	246
12.3.1	维护技术方面的工具	246
12.3.2	维护管理方面的工具	246
12.4	维护管理与文件	246
12.4.1	软件维护的管理	247
12.4.2	软件问题报告	248
12.4.3	软件修改报告	249
12.4.4	软件维护指南	251
12.5	小结	251
	思考与习题	252
第 13 章	软件质量管理	254
13.1	对软件质量的需求	254
13.1.1	用户的质量观	254
13.1.2	开发人员的质量观	254
13.1.3	维护人员的质量观	255
13.1.4	管理人员的质量观	255
13.2	软件质量度量	255
13.2.1	软件质量框架模型	255
13.2.2	软件质量特性	256
13.2.3	评估指标的选取原则	257
13.3	软件质量评估指标体系	258
13.3.1	功能度指标	258
13.3.2	可靠性指标	258
13.3.3	易用性指标	259
13.3.4	效率特征指标	260
13.4	软件评价	260
13.4.1	复杂性度量	261
13.4.2	可靠性评价	261
13.4.3	性能评价	262
13.4.4	运行评价	262
13.5	软件能力成熟度模型	264
13.6	质量管理文件	265

13.7 小结	266
思考与习题	266
第14章 软件管理	269
14.1 软件项目的特点与软件管理职能	269
14.1.1 软件项目的特点	269
14.1.2 软件管理的特殊困难	269
14.1.3 软件管理的主要职能	270
14.2 软件开发组织	270
14.2.1 软件设计小组的形式	270
14.2.2 对项目经理人员的要求	272
14.2.3 评价软件人员的条件	272
14.3 标准化管理	272
14.3.1 软件工程标准化	272
14.3.2 软件工程标准化的意义	273
14.3.3 软件工程国家标准	274
14.4 软件配置管理	276
14.4.1 软件配置	276
14.4.2 配置标识	277
14.4.3 里程碑与变更控制	278
14.4.4 配置状态登录	279
14.4.5 配置审查	279
14.4.6 计算机软件配置管理计划规范	280
14.5 小结	280
思考与习题	281
第15章 软件过程	282
15.1 生存周期过程与软件过程工程	282
15.1.1 软件生存周期过程	283
15.1.2 软件过程工程	284
15.1.3 软件过程工程和软件项目工程	285
15.2 生存周期基本过程	285
15.2.1 获取过程	285
15.2.2 供应过程	286
15.2.3 开发过程	287
15.2.4 运作过程	289
15.2.5 维护过程	289
15.3 生存周期支持过程	289
15.3.1 文档编制过程	290
15.3.2 配置管理过程	290

15.3.3	质量保证过程	290
15.3.4	验证过程	290
15.3.5	确认过程	291
15.3.6	联合评审过程	291
15.3.7	审核过程	291
15.3.8	问题解决过程	291
15.4	生存周期组织过程	291
15.4.1	管理过程	292
15.4.2	基础设施过程	292
15.4.3	改进过程	292
15.4.4	培训过程	292
15.5	剪裁过程	292
15.6	过程与组织的关系	293
15.7	Rational 统一过程	296
15.8	小结	298
	思考与习题	299
第 16 章	软件体系结构	300
16.1	软件体系结构	300
16.1.1	什么是体系结构	300
16.1.2	软件体系结构的定义	301
16.1.3	体系结构描述	303
16.1.4	体系结构决策	304
16.2	体系结构类型	305
16.3	体系结构风格	306
16.3.1	体系结构风格的简单分类	307
16.3.2	体系结构模式	309
16.3.3	组织和求精	310
16.4	体系结构设计	310
16.4.1	系统环境的表示	311
16.4.2	定义原型	312
16.4.3	将体系结构精化为构件	313
16.4.4	描述系统实例	314
16.5	评估可选的体系结构设计	314
16.5.1	体系结构权衡分析方法	315
16.5.2	体系结构复杂性	315
16.5.3	体系结构描述语言	316
16.6	小结	316
	思考与习题	317

第 17 章 软件工程的发展	318
17.1 原型化方法	318
17.2 软件复用和构件技术	319
17.2.1 软件复用的概念	319
17.2.2 软件复用的过程	320
17.2.3 构件和体系结构	320
17.2.4 构件和构件系统	320
17.3 敏捷开发	321
17.3.1 什么是敏捷	321
17.3.2 敏捷及变更的成本费用	322
17.3.3 敏捷过程	322
17.3.4 极限编程	324
17.4 软件工程的发展趋势	328
17.4.1 管理复杂性	329
17.4.2 开放世界的软件	330
17.4.3 意外需求	330
17.4.4 人才结构	331
17.4.5 软件构造块	331
17.4.6 对“价值”认识的转变	332
17.4.7 开源	332
17.5 软件工程的技术方向	332
17.5.1 巨大的挑战	333
17.5.2 协同开发	333
17.5.3 需求工程	334
17.5.4 模型驱动的软件开发	335
17.5.5 测试驱动的开发	335
17.6 小结	336
思考与习题	337
参考文献	338

第1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 20 世纪 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, 甚至更长时间。
- 3) 软件开发人员和用户之间信息交流不充分, 导致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。20 世纪 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学科的发展。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 计算机系统与软件

硬件工程和软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。今天, 硬件设计技术已经达到比较成熟的水平。目前, 制造方法仍在不断地改进, 可靠性已是一种可以期待的现实, 而不再是一种愿望。

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