



21世纪高职高专软件技术专业规划教材

RUANJIAN
GONGCHENG

软件工程

●主编 王 水

河南科学技术出版社

21 世纪高职高专软件技术专业规划教材

软 件 工 程

主 编 王 水

河南科学技术出版社
· 郑州 ·

图书在版编目(CIP)数据

软件工程/王水主编. —郑州:河南科学技术出版社,
2008.2

(21世纪高职高专软件技术专业规划教材)

ISBN 978-7-5349-3817-7

I. 软… II. 王… III. 软件工程-高等学校:技术学校-
教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 013661 号

出版发行:河南科学技术出版社

地址:郑州市经五路 66 号 邮编:450002

电话:(0371)65737028 65788613

网址:www.hnstp.cn

策划编辑:范广红

责任编辑:杨艳霞

责任校对:柯 姣

封面设计:张 伟

版式设计:南 妮

印 刷:河南第一新华印刷厂

经 销:全国新华书店

幅面尺寸:185mm×260mm 印张:17.5 字数:413 千字

版 次:2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

定 价:32.00 元

如发现印、装质量问题,影响阅读,请与出版社联系。



编写说明

BIANXIESHUOMING

近年来,我国国民经济快速发展,信息产业更是突飞猛进,再加上国际化软件技术专业人才的巨大市场需求,都为软件技术专业人才的培养提供了强大的驱动力。2001年12月,教育部与国家计划经济委员会联合提出了建立35所国家示范性软件学院的决定,并于2003年又批准35所高职高专院校为示范性软件职业技术学院。除此之外,各省又相继成立了一批软件学院或软件职业技术学院。目前,这些高校已成为高等职业教育的一个重要组成部分,培养了大批优秀的软件技术专业人才,为我国软件产业的可持续健康发展提供了强有力的支撑。

教材的建设对人才培养起着至关重要的作用。就如何做好课程体系的建设和人才培养工作,中国计算机学会教育委员会、高等学校计算机教育研究会联合组建的“中国计算机科学与技术学科教程2002研究组”推出的《中国计算机科学与技术学科教程2002》给出了16门核心课程描述,建议以此构建专业人才的公共平台;2006年,高等学校计算机教育研究会结合高职院校计算机教育的特点,又推出了《中国高职院校计算机教育课程体系》蓝皮书,给出了目前流行的计算机类各专业的参考课程体系架构。目前,国内各出版机构围绕这两本书出版了很多软件类教材,但是多偏重于编程语言理论的教学,大多为传统的教学模式,结果导致学生的编程设计能力和应用能力不够。为此,河南科学技术出版社组织郑州大学、河南大学、中原工学院、郑州轻工业学院、南阳理工学院、平顶山学院等省内软件技术学院教学一线的教师及软件开发公司具有丰富经验的工程技术人员共同编写了一批教材,共包括理论课教材20种,实训课教材13种。

本套教材以《中国计算机科学与技术学科教程2002》和《中国高职院校计算机教育课程体系》蓝皮书为

指导,以“就业为导向、能力为本位”要求为原则,以“淡化理论,强化能力,体现创新,灵活多用”为出发点,突出实际动手能力和实用性,突出案例和任务驱动等技能训练。为了培养外向型软件技术专业人才,还编写了《计算机专业英语》、《计算机专业日语》。

为满足不同学校、不同层次、不同基础水平进行教学安排和人才培养的实际需要,本套教材尽量采用富有弹性的模块化内容结构,对知识传授与能力培养采用有目的的综合、融合和综合的编写方法,将若干知识点组成模块,每个模块既是教材的有机组成部分,又是一个相对完整而开放的单元,以便于教师组织教学与学生自主学习。

同时,本套教材具有系列化、立体化特征,即在编写教材的同时,开发出一些好的电子课件,通过教学资源库、课程网站等供老师、学生使用。

本套教材既适合作软件学院、软件职业技术学院以及计算机相关专业的本、专科生教材,也可作为实训机构的培训教材和相关技术人员的学习参考书。

要编写一套推动和促进应用型人才培养的教材是一项艰巨的任务,加上软件技术专业的招生时间还比较短,可以借鉴的经验不多,尽管编审委员会与各位专家都已尽力,恐仍存在疏漏之处,恳请各位读者批评指正。

郑州大学软件技术学院 王世卿

2008年1月

单各会员委审



编写人员名单

BIANXIEBENYUAN MINGDAN

主 编 王 水

副主编 詹 英 王玉磊

编 委 (以姓氏笔画排序)

王 水 王玉磊 朱超武 杨 晶

张晓民 高 需 郭东恩 黄照鹤

詹 英

单 位 人 员



编审委员会名单

BIANSHEN WEIYUANHUI MING DAN

主 任 王世卿

副主任 车战斌 刘黎明 吴勇军 李 波

李占波 李 捷 张素智

委 员

(以姓氏笔画为序)

于立红 王世卿 车战斌 刘黎明

孙 杰 李 波 李 捷 李占波

吴勇军 张素智 陈桂生 秦国防

郭长庚



前言

QIANNAN

软件工程(Software Engineering,简称SE)是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。它涉及程序设计语言、数据库、软件开发工具、系统平台、标准、设计模式等方面。

高等院校教育的目标是培养具有一定基础理论、专业知识水平和较强实践操作技能的技术应用型人才。本教材在编写过程中强调内容的针对性,理论上强调必需、够用,技术上强调实用性。

本教材通过对软件工程各个方面的阐述,较全面地介绍了软件工程的基本概念、软件开发方法和开发工具,包括软件工程中软件的定义、设计、测试、维护,面向对象技术,软件复用技术,软件管理和质量保证等。试图向读者深入介绍软件工程的观念、意义和方法学。

本教材共分13章,大致可以分为四个部分,第一部分:软件工程基本概念,包括第1章至第2章。第二部分:软件工程的过程,包括第3章至第8章。第三部分:软件工程管理,包括第9章至第12章。第四部分:软件工程新趋势,包括第13章。

本教材适用于《软件工程》课程的教学用书,也可作为计算机软件从业人员的参考书。

本教材由南阳理工学院的王水、王玉磊、杨晶、张晓民、高需、郭东恩、黄照鹤、詹英和三门峡职业技术学院的朱超武编写,王水任主编,由王水(第1、第2章)、黄照鹤(第3、第4章)、杨晶(第5、第6章)、张晓民(第7章)、郭东恩(第8章)、朱超武(第9、第10章)、詹英(第11章)、高需(第12章)、王玉磊(第13章)具体参与编写相关章节。

时间仓促,难免出现错误,敬请读者指正。

编者

2007年12月

目录



M U L U

第1章 软件工程概述	1
1.1 软件的概念	1
1.1.1 软件的定义	1
1.1.2 软件的发展	2
1.1.3 软件的种类	3
1.1.4 软件产品的特征	4
1.2 软件危机	5
1.2.1 软件危机的含义	5
1.2.2 软件危机的表现	6
1.2.3 产生软件危机的根源	7
1.2.4 解决软件危机的途径	8
1.3 软件工程	10
1.3.1 软件工程的含义	10
1.3.2 软件工程的发展	13
1.3.3 软件工程的原则	15
1.3.4 软件工程的主要目标和环节	16
1.4 软件开发方法学	17
1.4.1 软件开发方法的发展	17
1.4.2 结构化分析与设计方法	18
1.4.3 面向对象分析与设计方法	20
1.5 软件工具与软件工程环境	22
1.5.1 软件工程辅助工具简介	22
1.5.2 软件工程环境	23
第2章 软件过程	26
2.1 软件过程概述	26
2.1.1 什么是软件过程	26

2.1.2	为什么我们需要一个软件过程	27
2.1.3	成熟度等级	27
2.1.4	关键过程领域(KPA)	29
2.2	软件生存周期	33
2.2.1	软件生命周期(SDLC)的六个阶段	34
2.2.2	软件生命周期模型	34
2.3	具体生命周期模型	35
2.3.1	瀑布模型	35
2.3.2	原型模型	36
2.3.3	增量模型	37
2.3.4	螺旋模型	39
2.3.5	构件集成模型	41
2.3.6	并发开发模型	42
2.3.7	第四代技术	43
第3章	软件需求分析	46
3.1	需求分析概述	46
3.1.1	需求分析的任务	46
3.1.2	需求分析的步骤	47
3.1.3	需求获取的常用方法	49
3.2	结构化分析方法	52
3.2.1	结构化分析概述	52
3.2.2	数据流图	52
3.2.3	数据词典	56
3.3	软件需求文档	58
第4章	软件设计概述	61
4.1	软件设计的任务	61
4.2	软件设计的基本概念	62
4.2.1	模块与构件	62
4.2.2	抽象与细化	62
4.2.3	信息隐藏与局部化	62
4.3	模块化设计	63
4.3.1	模块化设计产生的效果	63
4.3.2	模块分割方法	63
4.3.3	模块的独立性	64

4.3.4	模块设计规则	69
4.4	概要设计文档与评审	71
4.5	详细设计	71
4.5.1	详细设计概述	71
4.5.2	详细设计的方法	72
4.5.3	详细设计复审	74
第5章	面向对象的分析	76
5.1	面向对象的分析概述	76
5.1.1	面向对象分析的主要原则	77
5.1.2	面向对象分析产生三种分析模型	78
5.1.3	面向对象分析的主要优点	78
5.1.4	面向对象分析方法的基本步骤	78
5.2	功能模型(用例模型)	78
5.2.1	用例图	79
5.2.2	用例图元素之间的相互关系	81
5.2.3	寻找用例的方法	83
5.3	用例分析	83
5.3.1	用例分析概述	84
5.3.2	用例分析的步骤	84
第6章	面向对象的设计	89
6.1	面向对象的设计概述	89
6.1.1	面向对象的设计模式概念	90
6.1.2	面向对象设计的主要原则	90
6.1.3	设计模式分类	93
6.1.4	怎样才能设计“好的面向对象”	96
6.2	子系统设计	96
6.2.1	子系统设计概述	96
6.2.2	子系统设计的目的	97
6.2.3	子系统设计的步骤	97
6.3	类设计	98
6.3.1	类设计概述	98
6.3.2	类设计的目的	99
6.3.3	类设计的步骤	99
6.4	数据库设计	103

80	6.4.1 数据库设计概述	103
15	6.4.2 数据库设计的目的	105
15	6.4.3 数据库设计的步骤	105
15		
85	第7章 编码与实现	107
11	7.1 程序设计语言	107
	7.1.1 程序设计语言的发展	108
85	7.1.2 程序设计语言的分类及特点	112
85	7.1.3 程序设计语言选择	114
77	7.2 编码的目标和指导原则	115
85	7.3 编码的风格	116
85	7.3.1 程序版式	117
85	7.3.2 数据说明	118
85	7.3.3 语句结构	119
95	7.3.4 输入/输出	120
18	7.4 编程效率	121
88	7.5 编程的安全性	122
88	7.6 面向对象编程技术	123
48	7.6.1 面向对象编程概述	123
48	7.6.2 面向对象语言	124
	7.6.3 面向对象程序设计风格	126
98	7.6.4 面向对象实现基本准则	127
98	7.6.5 面向对象实现工作流程	128
00		
00	第8章 软件测试	130
80	8.1 软件测试概述	130
80	8.1.1 什么是软件测试	131
80	8.1.2 软件测试的目的	131
80	8.1.3 软件测试的原则	131
70	8.1.4 软件测试的方法	132
70	8.1.5 软件测试的基本过程	133
80	8.1.6 软件测试的文档	134
80	8.1.7 软件测试与调试的区别	136
90	8.1.8 测试团队构成及角色划分	136
90	8.2 白盒测试	138
801	8.2.1 什么是白盒测试	139

8.2.2	六种覆盖方法	139
8.2.3	基本路径测试	143
8.2.4	白盒测试工具的选择	145
8.2.5	白盒测试的优缺点	145
8.3	黑盒测试	146
8.3.1	什么是黑盒测试	146
8.3.2	黑盒测试的测试用例设计方法	146
8.3.3	黑盒测试工具的选择	149
8.3.4	黑盒测试的优缺点	149
8.4	软件测试级别	150
8.4.1	单元测试	150
8.4.2	集成测试	154
8.4.3	系统测试	159
8.4.4	回归测试	161
8.4.5	验收测试	164
8.5	测试过程和管理	167
8.5.1	软件测试过程模型介绍	168
8.5.2	测试过程管理理念	170
第9章	软件项目管理	173
9.1	软件项目管理基础	173
9.1.1	软件项目管理的意义	173
9.1.2	软件项目管理的定义及职能	174
9.1.3	软件项目管理的特点	174
9.1.4	软件项目管理的内容	175
9.1.5	提高软件成功的途径	176
9.2	软件项目计划	176
9.2.1	软件项目计划概念	176
9.2.2	软件项目计划内容	177
9.2.3	成本估算	178
9.2.4	进度安排	179
9.2.5	项目报告	181
9.2.6	项目检验	181
9.3	质量管理	182
9.3.1	软件质量保证	182
9.3.2	项目管理评审	184

9.4	项目中的组织	185
9.4.1	项目组织模式	185
9.4.2	人员配备	185
9.5	配置管理	186
9.5.1	基线	187
9.5.2	软件配置项	187
9.5.3	版本控制	188
9.5.4	变更控制及访问控制和同步控制	188
9.5.5	配置审计和状态报告	188
9.6	风险和风险管理	189
9.7	软件工程标准化与软件文档	190
9.7.1	软件工程标准化	190
9.7.2	软件工程标准化的意义	191
9.7.3	软件工程标准的层次	191
9.7.4	文档的作用与分类	192
第10章	软件质量保证	195
10.1	质量的概念	195
10.1.1	什么是质量	196
10.1.2	质量的演变	196
10.1.3	软件质量	197
10.1.4	软件质量的特性	198
10.2	质量框架	200
10.3	软件质量保证	201
10.4	软件质量度量	206
10.4.1	McCall 质量度量模型	206
10.4.2	ISO 的软件质量评价模型	207
10.4.3	软件质量度量	207
10.5	软件复杂性	208
10.5.1	软件复杂性的基本概念	208
10.5.2	软件复杂性的度量方法	209
10.6	软件可靠性	210
10.6.1	软件可靠性的定义	210
10.6.2	软件可靠性的指标	211
10.6.3	软件可靠性的模型	211
10.7	软件评审	212

10.7.1	设计质量的评审内容	212
10.7.2	程序质量的评审内容	213
10.8	软件容错技术	214
10.8.1	容错技术定义	214
10.8.2	容错的一般方法	215
10.8.3	容错软件的设计过程	216
10.9	软件开发各阶段的软件质量保证目标	216
10.9.1	需求分析	216
10.9.2	设计	216
10.9.3	编码	217
10.9.4	测试	217
10.9.5	维护	217
10.9.6	软件规范文档	217
10.10	质量保证与质量控制的区别	218
10.11	全面质量管理	219
10.11.1	计划—执行—检查—行动(PDCA)	220
10.11.2	质量改进范例	221
10.11.3	软件工程协会(Software Engineering Institute, 简称 SEI)	221
10.12	案例研究	222
第 11 章	软件配置管理	224
11.1	软件配置管理	225
11.1.1	基线	225
11.1.2	软件配置项	226
11.2	SCM 过程	229
11.3	软件配置中对象的标识	229
11.4	版本控制	231
11.5	变化控制	231
11.6	配置审计	232
11.7	状态报告	233
11.8	SCM 标准	233
第 12 章	软件维护	235
12.1	软件维护概述	235
12.1.1	软件维护和分类	235

12.1.2	软件维护的特点	236
12.1.3	软件维护策略	238
12.2	软件的可维护性	238
12.3	软件维护过程和模型	239
12.3.1	软件维护活动	239
12.3.2	软件维护的副作用	243
12.3.3	软件维护过程模型	244
12.4	软件再工程	246
第13章 软件工程新方法简介 248		
13.1	极限编程(Extreme Programming,简称XP)简介	248
13.1.1	极限编程的概念	249
13.1.2	XP的四条基本价值原则	250
13.1.3	XP的实践准则	250
13.1.4	XP的使用及学习	254
13.2	微软解决方案框架(Microsoft Solution Framework,简称MSF)简介	254
13.2.1	MSF的概念	255
13.2.2	MSF的八个基础原理	255
13.2.3	MSF模型	257
13.2.4	MSF准则	260
13.2.5	MSF的使用及学习	262
参考文献		263

第1章



软件工程概述

学习目标

本章主要介绍软件工程学中的基本概念、软件工程产生的历史背景、软件开发的基本方法以及软件工程辅助工具。

随着计算机应用的日益普及和广泛,计算机软件已经在整个计算机系统中占有越来越重要的地位。我们如何高效率地开发出高质量的软件,如何能让用户满意,怎样才能使软件更容易维护?这就是软件工程学所研究的问题。软件工程就是将各种工程的理论和方法应用于软件开发实践的科学。它借鉴传统工程的原理和方法,以求高效率地开发高质量的软件。它综合了计算机科学、数学、工程科学、管理科学等学科中与软件开发相关的理论和方法,形成一套适合于软件开发的方法、技术和规范。

1.1 软件的概念

20 世纪中叶,软件产业开始起步,在短短 50 年的时间里迅速发展成为推动人类社会发展的龙头产业,并造就了一批百万富翁、亿万富翁。随着信息产业的发展,软件产业在人类社会越来越重要。对于人类而言,软件是一个全新的东西,其发展历史不过四五十年。人们对软件的认识经历了一个由浅到深的过程。

1.1.1 软件的定义

在计算机系统发展的初期,硬件通常用来执行一个单一的程序,而这个程序又是为一个特定的目的而编制的。早期当通用硬件成为平常事情的时候,软件的通用性却是很