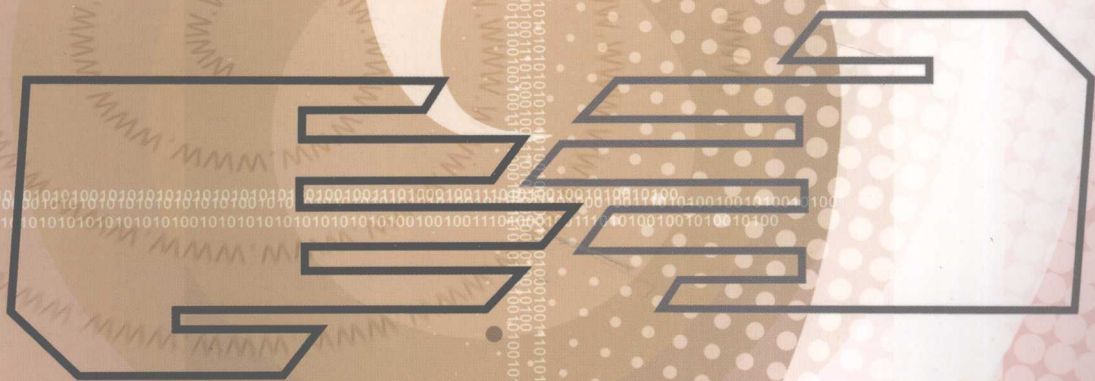


辽宁省高等教育自学考试指定教材（附自学考试大纲）

软件工程及应用

张斌 郭军 主编



东北大学出版社
Northeastern University Press

清华大学出版社

软件工程及应用

第二版



清华大学出版社

辽宁省高等教育自学考试指定教材(附自学考试大纲)

软件工程及应用

张 斌 郭 军 主 编

东北大学出版社

· 沈 阳 ·

内容提要

本书从应用型角度出发,以软件质量为中心,围绕软件工程中的过程和方法两大主题,系统介绍了软件工程的基本概念、原理和方法,内容包括软件过程的概念、软件过程模型、可行性分析和需求分析方法、软件设计方法、软件实施与测试方法以及软件项目管理方法等。每章后附有小结和习题,便于考生自学。

本书可作为应用型本科计算机软件专业自学考试指定教材,适用于普通高等学校计算机类或相关专业的教师和学生参考教材。

© 张 斌 郭 军 2007

图书在版编目 (CIP)

软件工程及应用 / 张斌, 郭军主编. — 沈阳: 东北大学出版社, 2007.5

ISBN 978-7-81102-385-5

I. 软… II. ①张… ②郭… III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2007) 第 061580 号

出版者: 东北大学出版社

地址: 沈阳市和平区文化路 3 号巷 11 号

邮编: 110004

电话: 024—83687331 (市场部) 83680267 (社务室)

传真: 024—83680180 (市场部) 83680265 (社务室)

E-mail: neuph@neupress.com

http://www.neupress.com

印刷者: 辽宁教育学院印刷厂

发行者: 东北大学出版社

幅面尺寸: 184mm×260mm

印 张: 13.75

字 数: 343 千字

出版时间: 2007 年 5 月第 1 版

印刷时间: 2007 年 5 月第 1 次印刷

责任编辑: 刘 莹

责任校对: 张 立

封面设计: 唐敏智

责任出版: 秦 力

ISBN 978-7-81102-385-5

定 价: 25.00 元

编审委员会

主任委员 李荣希

副主任委员 王新民 王殿元 马 强 张德君

于 健 李长江 夏 青

委 员 (以姓氏笔画为序)

于 健 马 强 尹久恒 王新民

王殿元 王新春 刘庆远 李荣希

李 光 李长江 卢俊杰 杨 东

苏显阳 张德君 徐大铨 夏 青

鞠绍岩

计算机软件专业（应用本科）教材 编写委员会

策 划 王殿元

主 任 张 斌

副主任 王兴伟 赵林亮 卢俊杰

委 员 （以姓氏笔画为序）

王兴伟 王元斌 王 纲 全 忠

卢俊杰 郭 军 张 斌 孟凡荣

林树宽 杨 雷 赵林亮 姚 羽

高 岩

组编前言

高等教育自学考试是个人自学、社会助学和国家考试相结合的高等教育形式，是我国高等教育体系的重要组成部分。高等教育自学考试应用本科属于自学考试范畴，是国家考试委员会、省自学考试委员会探索高等教育自学考试，培养具有较高文化素养的应用型、实用型、复合型人才的新模式。

开设高等教育自学考试计算机软件专业(应用本科)，旨在贯彻实施辽宁省教育厅《振兴辽宁老工业基地教育服务规划》，根据计算机应用在当前和今后一段时间内人才市场上的需求及我省企业的特点，引导社会自考生学有所用、服务社会、自学成才。本专业的总体要求与全日制普通高等学校相应或相近专业的本科水平相一致，并根据高等教育自学考试的特点，着重考核自考生对计算机软件技术的基础理论、软件开发与设计的基本技术、数据库和网络实用编程技术等实际技能的掌握程度，培养具有较高文化素养的社会急需的具有复合型特点的计算机软件实用型人才。

计算机软件专业(应用本科)是计算机学科中的一个新专业，与目前全国自学考试中的有关计算机应用类(独立本科段)的“计算机及应用”“计算机网络”和“计算机信息管理”等三个专业相比，有以下几个特点。

① 专业特点明显，培养目标明确。重点在于培养具有复合型特点的实用性软件技术人才。

② 从课程设置看，更加注重应用。专业基础课和专业课教材内容与上述专业相比有较大变化内容更新、更实用，更适应于市场需求。

③ 充分强调专业技能，增加实践学分，实践考核目标明确。

编写本系列教材的指导思想是，既遵循考试大纲要求，又充分体现自学考试特点，自成体系，理论够用并且实用。编写的依据是辽宁省自学考试计算机软件专业(应用本科)的考试计划，并参照教育部计算机科学与技术专业指导委员会最新下发的关于软件工程和信息技术方向的专业规范。

在东北大学出版社的大力支持和配合下，教材编写委员会组织有经

验和教学专长的教授及中青年骨干教师,陆续编写7门课程教学所需的理论与实践配套的14本教材,具体是:

1. 数据结构与算法
2. 数据结构与算法 (实践)
3. 计算机软件开发技术基础
4. 计算机软件开发技术基础 (实践)
5. 数据库技术及应用
6. 数据库技术及应用 (实践)
7. 面向网络编程技术
8. 面向网络编程技术 (实践)
9. 软件工程及应用
10. 软件工程及应用 (实践)
11. 软件测试技术
12. 软件测试技术 (实践)
13. 软件开发与设计实例分析
14. 软件开发与设计实例分析 (实践)

本系列教材的编写目标是:体现“教材就是为学生服务”的思想理念,使学生能够养成“理解—模仿—再理解—自觉应用”的良性循环学习方法,适合于考生自学;理论教材以讲解基本理论、基本方法和基本技术为重点;实践教材内容与理论教材内容同步,既反映专项技能训练,又体现综合应用能力。

希望使用本系列教材的师生不断地反馈各类意见,以便促进本系列教材的建设,为广大考生服好务。

计算机软件专业(应用本科)教材编写委员会

2007年4月

编者的话

自 20 世纪出现软件危机以来, 软件产业一直受到生产效率低下、产品质量不高、开发成本巨大、开发周期过长等问题的困扰。为了解决软件危机问题, Fritz Bauer 于 1968 年正式提出了软件工程的概念。经过几十年的努力, 软件产业经历了翻天覆地的变化, 人们总结了大量的经验, 发现了许多规律, 提出了很多方法, 并创造了大量工具。

在信息全球化大潮的推动下, 我国软件产业发展迅猛, 正在向软件大国的行列迈进。现代软件项目越来越复杂, 规模越来越大, 参与人员越来越多, 软件危机造成的影响也越来越大。在参与国际、国内市场的竞争中, 软件企业面临软件工程理念薄弱、过程管理粗糙、不善于运用软件工程方法和工具等问题, 迫切地需要软件专业人才能够很好地将软件工程运用到项目中, 摆脱或者缓解软件危机。计算机专业本科生作为软件企业未来的“主力军”, 能否在本科阶段掌握好软件工程, 打下坚实的理论基础, 并具有一定的实践能力, 具有十分重要的意义。根据自学考试课程大纲, 我们编写了本教材。

软件工程涉及软件过程、项目管理、开发方法、开发工具, 甚至企业文化等各个方面。我们从中选取了一些内容, 并努力使它们适合于本科生教学。本教材以软件质量为中心, 包括过程和方法两大主题, 并特别强调以下内容。

① 软件过程。第 2 章和第 3 章介绍了过程方面的知识, 包括过程概念、构成过程的一系列活动以及过程的组织策略(过程模型)。

② 可行性分析和需求分析方法。第 4 章和第 5 章从系统分析人员的视角出发, 讨论了项目可行性研究活动的任务、成本/效益分析的方法、需求分析的原则、传统的需求分析方法以及面向对象的需求分析方法。

③ 软件设计方法。第 6 章从软件设计人员的视角出发, 介绍了软件设计的总体原则、核心思想(分治、抽象、复用、内聚与耦合等)、传统的设计建模方法、面向对象的设计建模方法, 以及人机交互界面的设计。

④ 软件实施与测试方法。第 7 章从实施人员的视角出发, 讨论了软件开发过程中程序设计的语言和编码风格; 从测试人员的视角出发, 介绍了测试中的白盒测试、黑盒测试、单元测试和集成测试的方法。

⑤ 项目管理方法。第 8 章从项目管理人员的视角出发, 讨论了软件度量、软件计划和进度控制、人力资源管理、软件质量保证、软件风险管理和软件配置管理等方法。

本书第 1, 2, 5, 6, 7, 8 章由郭军编写; 第 3 章由高岩编写; 第 4 章由杨雷编写。张斌负责统筹全书并审核书稿。

尽管本教材努力吸取有关著作中的优秀内容, 但由于受到编者的水平、能力及

时间的限制，书中难免存在某些错误和不当之处，恳请专家、学者和广大读者不吝赐教。

为了更好地支持教学，我们还开发了软件工程课程网站 www.dcse.neu.edu.cn，提供课件、企业案例库和参考资料等教学素材的下载，以及网上答疑、在线题库等服务，欢迎访问。

本书在编写过程中，参阅了很多欧美著名软件工程专家，特别是 Roger S. Pressman, Hafeedh Mili, Timothy C. Lethbridge, Ivar Jacobson 和 Watts S. Humphrey 等人的著作，他们堪称经典的著作带给我们很多启迪。感谢东北大学软件工程课题组的所有成员，他们将多年来的教学经验无私地汇聚到这本教材中。感谢东北大学负责软件工程实验的指导教师，以及很多学习过软件工程课程的东北大学信息学院、软件学院和网络学院的学生们，他们在教学实践中提出了相当多的宝贵意见。

张 斌

2007年3月

目 录

第 1 章 软件工程概述	1
1.1 软件危机	1
1.2 软件工程概念	3
1.3 教材涉及的案例	5
习题.....	7
第 2 章 软件开发过程和活动	8
2.1 过程概念	8
2.2 问题定义活动.....	10
2.3 可行性研究活动.....	11
2.4 需求分析活动.....	15
2.5 设计活动.....	21
2.6 实施活动.....	27
2.7 测试活动.....	32
2.8 部署活动.....	37
习题	41
第 3 章 软件过程模型	43
3.1 过程模型概念.....	43
3.2 线形系列模型.....	44
3.2.1 线性顺序模型.....	44
3.2.2 瀑布模型.....	45
3.2.3 RAD 模型	47
3.3 演进系列模型.....	48
3.3.1 边建边改模型.....	48
3.3.2 增量模型.....	49
3.3.3 螺旋模型.....	50
3.3.4 RUP 模型	52
习题	55
第 4 章 问题定义和可行性研究方法	56
4.1 问题定义.....	56

4.2	可行性研究的任务	57
4.3	成本/效益分析	58
4.3.1	系统的启动成本与运营成本	58
4.3.2	系统的运营效益	59
4.3.3	成本/效益分析方法	60
4.3.4	成本/效益比较	61
4.4	可行性研究报告	62
4.4.1	可行性研究报告的内容	62
4.4.2	系统方案的内容和描述方法	64
4.4.3	词汇表	65
	习题	66
第5章	需求分析方法	67
5.1	需求分析的原则	67
5.1.1	循序渐进	67
5.1.2	自顶向下, 逐层分解	68
5.1.3	远离实现	69
5.1.4	可验证性	70
5.1.5	可追踪性	71
5.1.6	其他原则	72
5.2	传统需求分析建模方法	73
5.2.1	功能建模	74
5.2.2	行为建模	78
5.2.3	数据字典	80
5.2.4	判定表和判定树	83
5.3	面向对象的需求分析建模方法	85
5.3.1	用例建模	85
5.3.2	对象-关系模型	93
5.3.3	对象-行为模型	99
	习题	101
第6章	软件设计方法	103
6.1	设计的概念	103
6.1.1	设计的总体原则	103
6.1.2	软件构架设计	104
6.1.3	详细设计	105
6.2	分治	106
6.2.1	分治的优点	106
6.2.2	分治要考虑的问题	107

6.3 抽 象	109
6.3.1 抽象的优点以及基于抽象的设计原则	109
6.3.2 传统软件工程中的抽象	111
6.3.3 面向对象软件工程中的抽象	111
6.4 内聚与耦合	113
6.4.1 内聚的分类	113
6.4.2 耦合的分类	116
6.5 复 用	122
6.5.1 复用的概念	122
6.5.2 基于组件的软件复用	124
6.6 传统的设计建模方法	126
6.6.1 层次图	127
6.6.2 结构图	127
6.6.3 程序流程图	129
6.6.4 盒式图	129
6.6.5 PDL	130
6.7 面向对象的设计建模方法	130
6.7.1 包	131
6.7.2 类图	132
6.7.3 构件图	133
6.7.4 部署图	134
6.8 人机交互界面设计	135
6.8.1 以用户为中心的设计	135
6.8.2 界面设计指导原则	137
6.8.3 界面设计基础	138
习题	140
第7章 软件实施与测试方法	141
7.1 程序设计语言	141
7.1.1 广泛应用的程序设计语言	141
7.1.2 程序设计语言的选择	144
7.2 编码风格	145
7.2.1 程序的内部文档	145
7.2.2 程序的标识符	146
7.2.3 程序清单的安排	147
7.2.4 程序中的语句	148
7.3 软件测试原则	150
7.4 白盒测试	151
7.4.1 白盒测试的概念	151

7.4.2	基本途径测试	152
7.4.3	条件测试	154
7.4.4	循环测试	156
7.5	黑盒测试	157
7.5.1	黑盒测试的概念	157
7.5.2	等价类划分	157
7.5.3	边界值分析	159
7.6	单元测试	160
7.6.1	单元测试的考虑	160
7.6.2	单元测试的环境	161
7.7	集成测试	163
7.7.1	自顶向下集成测试	163
7.7.2	自底向上集成测试	164
7.7.3	自顶向下和自底向上相结合的集成测试	165
7.7.4	回归测试	165
	习题	166
第 8 章	软件项目管理方法	167
8.1	软件度量	167
8.1.1	软件特征量	167
8.1.2	软件开发成本和工作量的估计	170
8.2	软件计划和进度控制	171
8.2.1	软件计划	171
8.2.2	进度控制	173
8.3	人力资源管理	174
8.3.1	人力资源管理的任务	174
8.3.2	软件开发角色	175
8.4	软件质量保证	178
8.4.1	软件质量保证的概念	179
8.4.2	基于统计的软件质量保证	180
8.5	软件风险管理	181
8.6	软件配置管理	183
	习题	185
	模拟测试题	186
	参考答案	190
	参考文献	192
	附:《软件工程及应用》自学考试大纲	195



第1章 软件工程概述

为了更加有效地摆脱或者缓解软件开发过程中存在的周期长、成本高、质量低等问题，人们在20世纪60年代创立了一门工程学——软件工程。软件工程对软件产业的发展有着重要的意义。作为这门课程的开始，本章首先介绍软件工程的产生背景，即软件危机；然后描述软件工程的概概念；最后介绍一个贯穿在本教材中的项目案例。

1.1 软件危机

正如美国《商业周刊》在20世纪80年代初的一个头版大标题所说的“Software: The New Driving Force”，现在，政府、教育机构、金融行业、制造行业、媒体机构、航空科技、生命科学等处处运行着软件系统，软件已经融入我们所生活的这个世界的方方面面，成为人类发展的新的驱动力。随着人们越来越多地依赖于软件所带来的高性能计算和事务处理能力，软件系统也越来越复杂，程序量不再是一两张软盘那么小，系统从单机运行转变为网络上的分布式计算，Database由简单的DBASE发展成ORACLE，SYBASE这样的大型数据库，一个项目的开发从一个人“全拿”到多个人甚至多个团队的分工协作，专家系统、人工神经网络、并行计算、嵌入式系统等更智能、更复杂的运算被融入到软件中……

(1) 软件开发的特性

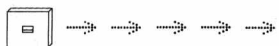
客户希望软件提供更多更好的服务，开发者希望能够更快更好地生产出软件。然而，在这些软件项目背后，却存在着很高的失败率。许多项目不是根本没有交付，就是推迟交付而且超出预算。即便是交付的软件系统，也常常会出现Bug太多而难以使用的问题。出现这些问题，是由于软件开发存在以下特性。

① 软件开发具有难以描述性。所采集的需求常常有多个来源，往往不完整，有时不正确，有时存在二义性，有时需求之间会有冲突。人们不能像描述一个化学反应那样运用一个明确的方程来描述软件的需求。

② 软件开发缺乏可见性。软件是“软”的、无形的，很难通过一行行的代码去度量其质量与进度。人们不能像观察一辆汽车的制造过程一样直观地观察一个软件的开发过程。不可见性造成开发团队及客户之间交流的困难，很可能导致最终产品存在重大的质量问题。

③ 软件开发具有复杂性。随着软件规模的增加，程序的复杂性呈指数上升。人们不能像小孩搭积木似的轻松地“搭造”出各式各样的软件产品，通过简单地更换部件来“改造”应用系统，因为这涉及技术能力、领域知识、分析设计方法、管理水平等诸多复杂因素，并且系统需求和质量标准也千差万别。此外，迄今为止，软件业还没有发展出足够丰富、足够简单、足够通用的“软件构件库”可供“搭造”。

④ 软件开发具有变化性。“变”似乎已经成为软件开发中的“常态”。客户需求随时会发生改变，如业务规则变化、软硬件环境变化（例如，从PC服务器升级到小型机、从MySQL数据库升级到ORACLE等）；开发人员对需求的理解程度、技术的掌握水平、人员



结构和项目的控制能力会持续变化；竞争的商业环境会发生变化；企业的市场策略会发生变化，等等。这些都会影响到项目的开发质量和开发周期。

⑤ 软件开发具有风险性。来自用户的、项目的、技术的和商业的风险随时随地都可能侵扰项目团队。人们不能像小孩堆沙那样随时可以推倒重来，因为软件开发不是沙滩游戏，一旦重来，客户和开发商都会付出很大代价。

⑥ 软件开发具有强合作性。随着软件规模越来越大、复杂程度越来越高，软件企业也越来越需要更多的人参与进来，以专业的水平完成不同角色的要求，大家分工协作，共同开发出质量更高而开发时间更短的产品。人们很难再像传统的裁缝店那样单枪匹马地生产产品。

(2) 软件危机

软件开发的这些特性使得人们深陷在软件危机中。所谓软件危机，是指在软件开发和维护过程中存在的周期长、成本高、质量低等问题。这些问题主要体现在以下方面。

① 缺乏计划性。软件开发计划（如进度安排和成本估计）一般很不准确，常常发生项目超时和成本超支现象。

② 软件需求获取不充分、不准确。不恰当的需求采集和分析方法使团队常常得到的是盲人摸象的结果，所开发的软件产品往往不能真正满足用户的实际需求。

③ 缺乏团队开发的合作、协调能力。手工作坊式的个体开发虽然能够达到“艺术化”的质量效果，但只适合小型的、简单的项目。对于大、中型软件项目来说，则需要按照一定的标准，由一个团队分工协作完成。

④ 缺乏良好的软件质量评测手段。它导致很难及时发现软件错误，软件质量常常得不到保证。

⑤ 软件的可维护性差。由于缺乏统一的开发规范，软件开发者的产品呈现“多样性”，造成产品难以被理解，更谈不上维护。缺乏完整的、规范的软件文档是软件可维护性差的另一个原因。

⑥ 软件的可复用性差。每次开发软件项目，常常是重新制造“车轮”。它导致大量的重复性劳动，极大地降低了软件开发效率。

⑦ 软件开发过程不规范。按照需求分析、设计、编码和维护这样一个简单的过程开发软件，常常导致项目大幅度返工甚至失败。有的项目团队甚至陷入“边建边改”的混乱过程模式中。

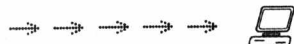
⑧ 缺乏自动化的工具来协助开发。进行建模和测试等开发工作时，若采用手工方式，则效率低下，满足不了团队快速开发的要求。

⑨ 缺乏变更管理措施。由于客户要修改需求、设计者要修改设计方案或者编码人员要修改代码等原因，软件项目常常发生变更。无序的变更常常导致项目变得混乱甚至崩溃。

⑩ 缺乏风险管理手段。随着软件项目规模越来越大、越来越复杂，如何识别、评估、应对和监控来自项目的、技术的和商业的各种风险，对软件企业来说，越来越重要。

“危机”一词本来是指某一段时期的危险、困难，但软件危机自出现以来，就没有消失过。大量的失败项目表明，软件危机并未远离我们，它已经伴随我们走过了几十年。

软件开发显然需要一个受控的过程，通过这个过程来集成软件开发的方方面面，并控制开发进度；软件开发需要一套方法去度量开发的质量，预测开发的周期和成本；软件开发需



要解决好资源（人力和设备）分配问题，企业应针对项目任务集合，提出良好的资源分配策略，以便达到最佳的团队合作效果；软件开发需要严格的开发标准来约束开发人员的创作思路，规范团队的工作“成果”，以构造工业化的“标准件”；软件开发需要采用“复用”思想，以减少重复开发带来的高成本和低效率……

逐渐地，人们意识到，要解决软件危机，必须采用“工程”的方法！如同建筑行业中的工程问题，软件开发涉及软件生产的各个方面，从最初的需求描述到最后的系统维护，从项目计划到资源调配，从开发标准到度量方法，等等，都属于工程范畴。软件开发人员必须运用科学的、系统的、有组织的工程方法，才能制作出高质量、低成本的软件产品。越是复杂的、大型的软件产品，越需要工程化的开发思想。

1.2 软件工程概念

软件工程是在软件危机背景下产生的。1967年，北大西洋公约组织（North Atlantic Treaty Organization，简称NATO）的计算机专家们针对软件危机的严峻形势，首次使用了“软件工程”这一术语，并于1968年在联邦德国召开的国际会议上，正式提出把在其他工程领域行之有效的一些工程学知识运用到软件开发过程中，从管理和技术两个方面，研究如何更好地开发与维护计算机软件的设想。软件工程学从此诞生。历史上，曾经有多位专家对“软件工程”这一概念给出了定义。

① Fritz Bauer在1968年NATO关于“软件危机”的会议上提出：“Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines”，他首次提出需要运用工程原理来实现在软件开发中高性价比的目标。这是人们提出的最早的“软件工程”概念。

② IEEE提出了一个更进一步的定义：a) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. b) The study of approaches as in a).

③ 加拿大标准协会提出：软件工程是软件设计、实现和测试过程中的系统化的活动，用于优化软件的生产与支持。

④ 英国著名软件工程学家 Ian Sommerville认为：Software engineering is an engineering discipline which is concerned with all aspects of software production.

⑤ 加拿大的 Timothy C. Lethbridge 等将软件工程定义为：The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints.

总结以上定义，并根据当前软件工程的发展特点，可以对软件工程给出以下定义：

软件工程是以质量为核心，为了经济地开发满足客户需求的软件而研究、建立和应用的系统化的、有规则的、可度量的和可控制的工程原则、方法，涉及软件过程、项目管理、开发方法、开发工具，甚至企业文化等各个方面。

分析这个定义，考察其中每个组成部分。

① 以质量为中心。软件工程的核心目标是保证和提高软件质量。Ralston 等人《Encyclopedia of Computer Science》一书中说：“软件质量保证应贯穿于整个开发过程。”所