



高等职业院校

技能型紧缺人才培养培训系列教材

软件工程初步

(计算机应用与软件技术专业)

主编 贾长云 副主编 朱敏 陆卫忠



高等教育出版社

高等职业院校

技能型紧缺人才培养培训系列教材

软件工程初步

(计算机应用与软件技术专业)

主 编 贾长云

副主编 朱 敏 陆卫忠

高等教育出版社

内容提要

本书根据《高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》的要求而编写。

本书针对职业院校学生的特点,充分体现《指导方案》中素质为基础、能力为本位、需求为依据、就业为导向的基本原则,教学内容注重选择当前软件工程的一些新理论、新方法与新技术,通过一个小型软件项目为案例贯穿全书,使学生在学完本书后能掌握软件工程的基本理论与方法以及它们在软件项目开发中的实际应用。其主要内容涉及软件工程中的软件系统策划、需求分析、软件设计、编码实现、软件测试及实施与维护等方面的内容,考虑到职业院校学生的就业定位,重点侧重于详细设计、软件编码、软件测试、软件维护这几个方面的基本知识与技能。为保持教学内容的先进性,同时便于对教学进行科学、灵活的组织,对诸如 UML 方法、面向对象的分析、面向对象的设计与编程、面向对象软件测试、CMM 等也做了介绍。

本书可以作为职业院校计算机类专业的软件工程课程的教材或参考书,也可作为 IT 行业程序员、测试员、维护员等的培训教材或参考书。

图书在版编目(CIP)数据

软件工程初步/贾长云主编. —北京:高等教育出版社,2004.7

ISBN 7-04-015161-8

I. 软... II. 贾... 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(2004)第 054790 号

策划编辑	李 波	责任编辑	张海波	封面设计	刘晓翔	责任绘图	宗小梅
版式设计	范晓红	责任校对	康晓燕	责任印制	孔 源		

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-82028899

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 北京乾沣印刷有限公司

开 本 787×1092 1/16
印 张 14.5
字 数 350 000

版 次 2004 年 7 月第 1 版
印 次 2004 年 7 月第 1 次印刷
定 价 18.10 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

出版说明

为了贯彻《国务院关于推进职业教育改革与发展的决定》的精神,促进职业教育更好地适应社会主义现代化建设对生产、服务第一线技能型人才的需要,教育部、劳动和社会保障部、国防科工委、信息产业部、交通部、卫生部联合发出了关于实施“职业院校制造业和现代服务业技能型紧缺人才培养培训工程”的通知。

根据“工程”的精神,教育部、信息产业部联合推出了《高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》,对职业教育教学改革提出了新的要求。即:职业教育是就业教育,要按照职业教育本身所固有的规律,在借鉴国内外成功经验的基础上,建立具有鲜明职业教育特点的课程体系。方案强调照顾学生的经验,强调合作与交流,强调多种教学方式交替使用,强调教师是学生学习过程的组织和对话伙伴。

为了帮助职业学校教师理解新的教学理念,更好地实施技能型紧缺人才培养计划,在深刻了解新的教学指导方案的基础上,高等教育出版社率先出版一套计算机应用与软件技术专业领域教材,以期帮助教师理解方案和组织教学,其特点有:

1. 借鉴国外先进的职业教育经验

研究了国外职业教育的各种模式,如英国的 BTEC 模式,印度的 NIIT 模式,澳大利亚的 TAFE 模式等,学习借鉴这些模式的优秀之处,又不拘泥于某种模式。

2. 协作式学习方式

强调以学生的团队学习为主,学生分成小组共同就某些问题进行讨论。同时认为学习与思考同等重要。在有限的时间内,使学生最大限度地掌握技能,并掌握自主学习的方法,为其今后的知识和能力拓展打下良好的基础。通过这种方法,有效地培养学生的沟通能力,如口头表达能力、书面表达能力、理解他人的能力和发表自己见解的能力。

3. 采用项目教学法组织教材

通过项目的活动过程培养学生的分析问题能力,团队精神,法律意识,沟通能力。每个项目相对较小,使学生对单个项目的学习过程不会太长,以减少学生的学习难度,提高学习兴趣。

4. 精心组织教材开发队伍

邀请教育专家、计算机专家、企业人士、职教教师共同参与项目开发,特别注意吸收双师型教师参加。

5. 根据项目特点设计课程解决方案

教材的组织是一个项目的解决方案,不是知识的细化,不以教会学生知识为目标,而以帮助学生掌握项目实施过程为目的。

6. 提供分层教学

书中实训指导、作业编排有一定梯度,以适应不同类别、不同能力学生的需要。

7. 配套完备的教学解决方案

教材出版的同时,与之配套的电子教案及与教材相关的素材将通过 <http://sv.hep.com.cn> 公布,供任课教师免费下载。

通过以上方式,高等教育出版社将为职业院校师生提供精良的教学服务,有不完备的地方也欢迎广大的职业院校的师生给予批评指正。

高等教育出版社

2004 年 5 月

序

致力于软件乃至计算机系统的研究、设计和实现有很多年了,在这种实践中逐渐认识到软件工程作为一种方法学的重要性。我于1995年在美国密歇根大学(University of Michigan)、1996年在英国城市大学(City University)较为系统地研究了设计方法学(Design Method),感悟到我们的专业教学似乎过多地注重对象本身,而忽略了解决问题的方法,因而,长时期地难以“纲举目张”。那以后,我开始在本科生和研究生中主讲软件工程课程,借助项目(project)的渗透和互补,不断地有新的领悟和进取。

有机会看到贾长云等几位年轻教师撰写的《软件工程初步》一书,刚一浏览,就为他们能把内容纷繁、易流于空泛、难入其境的软件工程学,通过生命周期(life cycle)这条脉络,将思想、方法和技术非常直观、具体地展现出来而感到振奋。全书以“教师教学网络测评系统”为案例,贯穿几乎所有章节,前后呼应,顺理成章。我有幸先作学生,从中能够感受并触摸到软件工程作为软件开发和管理方法学的具体意义和应用价值。

软件工程倡导的分析、设计、实现和维护的完整生命链,以及伴随着严谨的项目和过程管理,使软件摆脱了手工业者的思维定势和作坊式的劳作,虽还谈不上步入了工业化生产时期,但已成为摆脱软件神话和克服软件危机的一条必由之路。从强调前端(front-end)分析和后端(back-end)测试,淡化中间编程的40-20-40规则,到注重团队协作、服从系统统一、遵循配置管理等一系列原则,都说明软件工程真正从哲学、方法学、技术科学等多元、多层次的角度让参与软件投资、开发、使用 and 管理的各方都能体会到它的作用,告诉大家,软件决不仅仅是那些曾经令人觉得高深莫测的神秘代码(code)。机器指令也罢,高级程序设计语言也罢,都只是一种表达方式。正如作者在书中所述的那样,难以想像,只是依靠个体,哪怕你是一个“编程狂人”(coding fan),能实现像Windows XP这样具备4 000万行源代码、耗资50亿美金的软件项目。《软件工程初步》一书章章节节中都体现了这些思想,读到此书的人应该会从中获得裨益。

陈金水

2004年3月16日于河海大学

前 言

为了快速培养面向 21 世纪技能型紧缺人才,教育部、劳动与社会保障部等 6 部委于 2004 年初联合启动了“技能型紧缺人才培养培训”工程,计算机应用和软件技术专业列入了培养工程的重点专业。本书正是为了配合该工程,面向职业院校的计算机应用和软件专业学生而专门编写的。

软件工程学科自 20 世纪 60 年代末诞生以来,经过 30 多年的飞速发展,无论是理论还是实践都日趋成熟,现已经发展成为计算机科学与技术领域中的一项重要学科。软件工程是研究软件开发和软件管理的工程科学,是计算机应用及软件技术专业的主干专业课,也是软件分析设计人员、程序开发人员、软件测试人员、软件管理人员与营销人员、软件维护人员及软件企业的高层决策者都必须了解并掌握的不可缺少的专门知识。

目前,绝大部分与计算机应用和软件技术专业相关的学校,上至研究生,下至中等职业学校的学生都要开设软件工程相关课程。但学生尤其是职业院校的学生对学习软件工程课程的积极性不是很高,凡是担任过软件工程教学的老师大多都有过这种经历:老师在课堂上口若悬河,而学生听课时昏昏欲睡,课后学生对内容不知所以,考试时只求混个 60 分万事大吉。这种现象在职业院校的学生中特别严重。何以至此?原因有三:第一,软件工程本身的理论性、概念性相当强,职业教育的学生不容易接受;第二,我国很多软件企业都是小作坊式的,在其软件开发过程中根本就不考虑也不可能采用软件工程的思想与方法,因而学生感到软件工程没用;第三,现在许多软件工程教材只注重理论介绍,而不考虑实际应用,通篇条条框框,满眼基本概念,没有联系实际,更无案例介绍,这种教材对职业院校的学生来说如同天书,学习积极性与学习效果可想而知。

因此,本书在编写时充分体现了技能型人才培养的 4 个原则。

1. 以全面素质为基础,以能力为本位

提高素质、培养能力是本书编写时的第一原则,全书以一个小型软件系统作为典型案例,从软件定义、需求分析到软件设计、软件测试,直到软件的发布与实施无一不与案例相联系,使学生学完后,既能掌握软件工程的一般理论与方法,又能对软件的完整开发过程与管理过程以及相应的文档有一个完整、全面的了解。同时为了强化学生的课后学习,全书还特别设计了一个要求学生完成的小型软件系统贯穿各章。

2. 以企业需求为基本依据,以就业为导向

职业院校学生的就业岗位一般是软件企业中的程序开发人员、软件测试人员、营销人员及维护人员,因此本书在编写时既考虑到软件工程的完整体系结构,同时又充分考虑了目前软件企业对职业院校计算机类学生的基本需求,在软件实现与编码、软件测试、软件实施与维护上重点突出,而对于软件定义、需求分析、软件设计、软件管理等方面的内容不做过高要求。

3. 适应行业技术发展,体现教学内容的先进性和前瞻性

软件工程学科与其他计算机科学一样发展是非常迅速的,教材中对软件工程当前在实际应用中的新内容与新方法做了重点介绍,如 UML 方法、面向对象的分析、面向对象的设计与编码、面向对象软件测试等。对目前非常流行的 CMM 也做了简单介绍。

4. 以学生为主体,体现教学组织的科学性和灵活性

本书编写时强调内容的编排与篇章结构与学生的思维认知规律相统一,全书可分为软件开发过程与软件管理过程两大部分。对职业院校的学生来说,教学的重点应该在软件的开发过程。而本书对软件的管理过程的介绍则偏重于基本概念与基本方法。这样有利于各个学校在讲授时进行适当的增减,体现了教学组织的科学性与灵活性。

全书共分为 9 章,第 1 章介绍了软件工程中的基本概念与基本原理,也可说它是全书的一个缩影;第 2 章至第 7 章按照软件生存周期的各个阶段分别介绍了系统策划、需求分析、软件设计、编码、软件测试、软件的实施与维护,通过这 6 章将软件开发的完整过程呈现在学生面前;第 8 章和第 9 章则偏重于管理过程,包括软件项目管理及软件过程管理。在每一章后还设计了内容丰富、有一定层次性的习题,基本概念题比较简单,而综合分析题则有一定难度,供学习时选做。全书建议学时为 48~64 学时,其前导课程为程序设计语言、面向对象程序设计、数据库原理与应用。

本书第 1、6 章和第 4、8、9 章分别由淮海工学院贾长云和朱敏编写,第 2、7 章由苏州科技学院陆卫忠编写,第 3、5 章由湖北信息工程学校邬天菊编写。全书由贾长云负责统稿,朱敏、陆卫忠任副主编,软件工程专家河海大学博士生导师陈金水教授在百忙之中对本书进行了审阅,并提出了宝贵的意见。

在本书的编写过程中得到了许多单位及朋友的大力协助,连云港市科技局的王祥高级工程师、山东电子工业学校的王协瑞高级讲师等为本书提供了部分资料;江苏省自动化研究所、山东浪潮软件公司也为本书提供了部分软件文档,在此一并致谢。

本书可以作为职业院校计算机类专业的软件工程课程的教材或教学参考书,也可作为 IT 行业程序员、测试员、维护员等的培训教材或参考指南。为了便于组织教学和学习,本书的电子教案、案例等相关的完整文档、国家标准软件工程文档模板均可在 <http://sv.hep.com.cn> 上下载。

由于作者知识的局限,再加上编写时间仓促,书中的错误和缺陷就像软件中的错误与缺陷一样在所难免,欢迎广大读者对待本书如同测试员对待软件那样严格测试并找出错误,提出批评,更希望能附带改进意见,作者将不胜感谢,这样也能促使本书有进一步的提高。作者联系方式: E-mail: lyghhitjcy@vip.sina.com。

作 者

2004 年 3 月于连云港

目 录

第 1 章 软件工程概述	1
1.1 软件及软件工程	1
1.1.1 软件	1
1.1.2 软件工程	4
1.2 软件生存周期与开发模型	8
1.2.1 软件生存周期	8
1.2.2 软件开发模型	10
1.3 软件工程过程	14
1.3.1 软件工程过程的概念	14
1.3.2 常用软件过程管理 方法简介	14
1.4 软件开发工具简介	15
1.4.1 CASE 工具的 作用与分类	16
1.4.2 常用 CASE 工具简介	18
1.5 学习指南	20
1.5.1 本书案例	20
1.5.2 案例文档索引	22
1.5.3 习题说明	22
习题	23
第 2 章 系统策划	24
2.1 可行性研究	24
2.1.1 问题的定义	24
2.1.2 可行性研究的 任务与步骤	26
2.1.3 技术可行性分析	27
2.1.4 经济可行性分析	28
2.1.5 案例分析——可 行性研究	30
2.2 软件项目计划	32
2.2.1 软件项目计划的 主要内容	32
2.2.2 案例分析——软件 项目开发计划书	35
习题	36
第 3 章 需求分析	38
3.1 需求分析概述	38
3.1.1 需求分析的重要性	38
3.1.2 需求分析的任务	39
3.1.3 需求分析的过程与 方法	39
3.2 结构化分析	41
3.2.1 数据字典	42
3.2.2 数据流图	44
3.2.3 状态转换图	45
3.2.4 需求规格说明书	46
3.3 面向对象的分析	47
3.3.1 面向对象的概念	48
3.3.2 面向对象方法简介	50
3.3.3 面向对象分析过程	51
3.4 UML 概述	53
3.4.1 UML 的结构	53
3.4.2 UML 的图	54
3.4.3 UML 的应用	55
3.4.4 UML 中的需求分析 ——用例图的创建	57
习题	60
第 4 章 软件设计	61
4.1 软件设计基本概念	61
4.1.1 软件设计	61

4.1.2 软件模块	63	及典型案例	110
4.2 概要设计	64	5.6 程序员的基本素质要求	116
4.2.1 设计程序的模块结构	65	习题	117
4.2.2 设计程序的数据结构	66	第6章 软件测试	119
4.3 详细设计	67	6.1 软件测试概述	119
4.3.1 详细设计的表示	67	6.1.1 软件缺陷典型 案例分析	119
4.3.2 详细设计方法 (Jackson 方法)	70	6.1.2 软件测试的基本概念	120
4.4 面向对象的分析与设计	73	6.1.3 软件测试的 目标和原则	121
4.4.1 静态建模	73	6.2 测试技术分类及测试用例	123
4.4.2 动态建模	83	6.2.1 测试技术分类	123
习题	87	6.2.2 测试用例	124
第5章 编码	89	6.3 黑盒测试及其测试用例设计	126
5.1 程序设计语言	89	6.3.1 等价分类法	126
5.1.1 常用程序设计语言 及其特点	89	6.3.2 边界值分析法	128
5.1.2 程序设计语言的选择	95	6.3.3 错误推测法	129
5.2 编码规范	96	6.4 白盒测试及其测试用例设计	129
5.2.1 代码文档化	96	6.4.1 静态白盒分析 ——代码审查	129
5.2.2 数据说明与语句	97	6.4.2 动态白盒测试	131
5.2.3 输入/输出	97	6.5 软件测试策略	138
5.2.4 程序布局	98	6.5.1 测试流程与组织	138
5.2.5 注释	99	6.5.2 测试计划	141
5.3 结构化程序设计	100	6.5.3 单元测试	143
5.3.1 结构化程序 设计的原则	100	6.5.4 集成测试	145
5.3.2 结构化程序 设计的方法	100	6.5.5 确认测试	147
5.4 面向对象的程序设计	103	6.5.6 系统测试	148
5.4.1 面向对象的程序 设计语言	103	6.5.7 测试分析报告	149
5.4.2 面向对象程序设计 语言的设计风格	104	6.6 面向对象的软件测试	151
5.5 用户界面设计	107	6.6.1 面向对象技术对 传统测试的影响	151
5.5.1 人机界面设计的 一般问题	107	6.6.2 面向对象的测试 策略与步骤	152
5.5.2 人机界面设计过程	109	6.7 程序调试	155
5.5.3 人机界面设计实现原则		6.7.1 程序调试技术	155
		6.7.2 程序调试策略	156
		习题	157

第7章 软件实施与维护	159	8.1 软件项目计划与组织	178
7.1 软件用户文档	159	8.2 软件项目成本管理	180
7.1.1 软件文档	159	8.2.1 资源计划	181
7.1.2 用户操作手册及编制	160	8.2.2 成本估算、预算与控制	181
7.2 软件产品的发布与实施	163	8.3 软件项目进度控制	184
7.2.1 软件产品的发布	163	8.4 软件质量保证	186
7.2.2 软件产品实施过程	164	8.4.1 软件质量因素	187
7.3 软件维护的基本概念	165	8.4.2 软件质量保证 体系与实施	187
7.3.1 软件维护的概念	165	8.5 软件配置管理	190
7.3.2 软件维护的种类	165	8.5.1 配置管理任务	191
7.3.3 软件维护的代价	166	8.5.2 配置管理工具	194
7.4 软件维护的策略及副作用	167	习题	196
7.4.1 软件维护策略	167	第9章 软件过程管理	197
7.4.2 程序修改	171	9.1 软件能力成熟度模型 CMM	197
7.4.3 软件维护的副作用	173	9.1.1 CMM 的产生	197
7.5 软件维护中的新问题	174	9.1.2 CMM 内容简介	199
7.5.1 软件结构对 维护的影响	174	9.1.3 CMM 的应用	207
7.5.2 因特网对软件 维护的影响	175	9.2 个体软件过程 PSP	208
7.5.3 UML 对软件 维护的影响	176	9.3 统一过程 RUP	211
7.5.4 CMM 对软件 维护的影响	176	9.3.1 软件生存周期中的 各个阶段	213
习题	176	9.3.2 RUP 的核心 workflow	214
第8章 软件项目管理	178	习题	215
		参考文献	216

第 1 章

软件工程概述

随着计算机应用日益普及和深化,计算机软件的数量以惊人的速度急剧膨胀,而且现代软件的规模往往十分庞大,包含数百万行代码、耗资几十亿美元、花费几千人年的劳动才开发出来的软件产品,现在已经屡见不鲜了。例如,Windows 3.1 约有 250 万行代码,而现在被广泛使用的 Windows XP 的开发历时 3 年,代码约有 4 000 万行,耗资 50 亿美元,仅产品促销就花费了 2.5 亿美元。为了降低软件开发的成本,提高软件的开发效率,20 世纪 60 年代末诞生了一门新的工程学科——软件工程学。

1.1 软件及软件工程

1.1.1 软件

现在绝大部分人都知道计算机系统由软件和硬件两大部分构成,但对软件的含义很多人可能还不是很清楚。

软件的定义是随着计算机技术的发展而逐步完善的。在 20 世纪 50 年代,人们认为软件就是程序;60 年代人们认识到软件的开发文档在软件中的重要作用,提出软件等于程序加文档,但这里的文档仅指软件开发过程中所涉及的分析、设计、实现、测试、维护等,不包括管理文档;到了 70 年代人们又给软件的定义中加入了数据。因此,软件是计算机系统中与硬件相互依存的一部分,它包括以下 3 部分。

- 在运行中能提供所希望的功能与性能的程序。
- 使程序能够正确运行的数据及其结构。
- 描述软件研制过程和方法的所有文档。

1. 软件的特点

从广义来说,软件与硬件一样也是产品,但两者之间是有差别的,了解这种差别对理解软件工程是非常重要的。

(1) 软件角色的双重特性

软件作为一种产品具有双重特性,一方面它是一个产品,利用它来表现计算机硬件的计算潜能。无论是在主机中,还是驻留在设备(如手机)中,软件就是一个信息转换器,可以产生、管理、获取、修改、显示或传送信息。而另一方面它又是产品交付使用的载体,它可以控制计算机(如操作系统),可以实现计算机之间的通信,又可以创建其他程序。

(2) 软件的开发过程不同于硬件的制造过程

一般意义上的产品包括计算机硬件产品总要经过分析、设计、制造、测试等过程,也就是说要经过一个从无形的设想到一个有形的产品的过程。但软件仅仅是一个逻辑上的产品而不是有形的元件,软件是通过人的智力劳动设计开发出来的,而不是制造出来的。而且软件一旦被开发出来,就可以方便地进行大量的复制,因此其研制成本要远远大于生产成本。这也意味着软件的开发不能像制造产品那样进行管理。

(3) 软件不会“磨损”,但会退化

一般情况下,有形的硬件产品在使用过程中总会要磨损。在使用初期,往往磨损比较严重(这实际上是磨合),而经过短暂的磨合后,产品将进入相对的稳定期。由于任何硬件产品总有一定的生存周期,随着时间的流逝,硬件的各组成部分将出现各种各样、不同程度的磨损,这时硬

件的故障率必然大大提高,这也意味着硬件的寿命快要到了。硬件故障率与时间的关系可以用图 1-1 所示的“浴缸曲线”来表示。

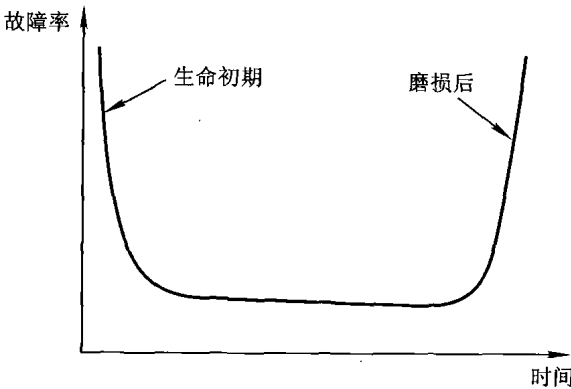


图 1-1 硬件故障率曲线

但对于软件来说,由于软件并不是一种有形的产品,因此也就不存在所谓的“磨损”问题。理想情况下,软件的故障曲线应该是图 1-2 中所示的理想曲线。在软件的运行初期,由于未知的错误会引起程序在其生命初期有较高的故障率,然而当修正了这些错误而且也没有引入新的错误后,软件将进入一种比较理想的平稳运行期。这说明软件是不会“磨损”的。但在实际情况中,软件尽管不会“磨损”,却会退化,如图 1-2 中的实际曲线。这是因为软件在其生存周期中会经历多次修改,每次修改都会引入新的错误,而对这些错误又要进行新的修改,使得软件的故障曲线呈现锯齿形,导致故障率慢慢升高,即软件产生了退化,而这种退化缘于修改。

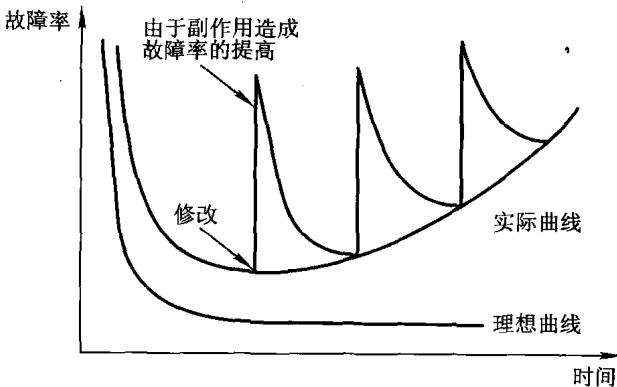


图 1-2 软件故障率曲线

(4) 绝大多数软件都是定制的且是手工完成的

在硬件制造业,构件的复用是非常普遍的。但由于软件本身的特殊性,构件复用才刚刚起步。理想情况下软件构件应该被设计成能够被复用于不同的程序,尽管今天的面向对象技术、构件技术已经使软件的复用逐渐流行,但这种复用还不能做到像硬件产品那样拿来即用,还需要进行必要的定制(构件之间的组合、接口的设计、功能的修改与扩充等),而且软件开发中构件的使用比例也是有限的。整个软件产品的设计基本上还是依赖于人们的智力与手工劳动。

(5) 开发过程复杂且费用昂贵

现代软件的体系结构越来越复杂,规模越来越庞大,所涉及的学科也越来越多,导致了软件的开发过程也异常复杂。靠一个人单枪匹马开发一套软件的时代已经一去不复返了,软件的开发需要一个分工明确、层次合理、组织严密的团队才能完成,显然软件的开发成本也会越来越昂贵。

2. 软件的分类

软件的应用非常广泛,几乎渗透到各行各业,因此要给出关于计算机软件一个科学的、统一的、严格的分类标准是不现实也是不可能的,但可以从不同的角度对软件进行适当的分类。常用的分类方法如表 1-1 所示。

表 1-1 软件的分类

分类序号	分类方法	对应类别	典型应用与特征
1	按功能分类	(1)系统软件	与计算机硬件的接口并为其其他程序服务,如操作系统、驱动程序等
		(2)支撑软件	用于开发软件的工具性软件,如开发平台、数据库管理系统、各种工具软件等
		(3)应用软件	为解决某一领域的问题而开发的软件,如商业软件、嵌入式软件、财务软件、办公软件、绘图软件等
2	按版权分类	(1)商业软件	版权受法律保护、经授权方可使用且必须购买的软件
		(2)共享软件	与商业软件类似,但可以先免费使用其试用版,其获取途径主要是通过因特网
		(3)自由(免费)软件	无须支付许可证费用便可得到和使用的软件,获取途径类似于共享软件
		(4)公有领域软件	没有版权,任何人都可以使用而且可以获得源代码的软件
3	按工作方式分类	(1)实时软件	用于及时处理实时发生的事件的软件,如控制系统、订票系统软件等
		(2)分时软件	允许多个联机用户同时使用计算机的软件
		(3)交互式软件	能够实现人机通信的软件
		(4)批处理软件	多个作业或多批数据一次运行、顺序处理的软件
4	按销售方式分类	(1)定制软件	受某个特定的客户委托,在合同的约束下开发的软件
		(2)产品软件	由软件开发机构开发可以为众多用户服务的,并直接提供给市场的软件

1.1.2 软件工程

1. 软件危机与软件神话

随着微电子技术的进步,计算机硬件性价比平均每 10 年提高 2 个数量级,而且质量稳步提高;与此同时,计算机软件成本却在逐年上升且质量没有可靠的保证,软件开发的生产率也远远跟不上计算机应用的需求。可以说软件已经成为限制计算机系统发展的关键因素。在 20 世纪 60~70 年代,西方计算机科学家把软件开发和维护过程中遇到的一系列严重问题统称为“软件危机”,它表现为以下几方面。

- 软件开发的生产率远远不能满足客观需要,使得人们不能充分利用现代计算机硬件所提供的巨大潜力。
- 开发的软件产品往往与用户的实际需要相差甚远,软件开发过程中不能很好地了解并理解用户的需求,也不能适应用户需求的变化。
- 软件产品质量与可维护性差,软件的质量管理没有贯穿到软件开发的全过程,直接导致所提交的软件存在很多难以改正的错误。软件的开发基本没有实现软件的可重用,它不能适应硬件环境的变化,也很难在原有软件中增加一些新的功能。加之软件的文档资料通常既不完整也不合格,使得软件的维护变得非常困难。
- 软件开发的进度计划与成本的估计很不准确,实际成本可能会比估计成本高出一个数量级,而实际进度却比计划进度延迟几个月甚至几年。开发商为了赶进度或节约成本会采取一些权宜之计,这往往会使软件的质量大大降低。这些现象极大地损害了软件开发商的信誉。

由上述现象可以看出,所谓的“软件危机”并不仅仅表现在不能开发出完成预定功能的软件,更大的问题是如何开发软件、如何维护大量已经存在的软件以及开发速度如何跟上目前对软件越来越多的需求。而造成这些现象的主要原因可以追溯到软件开发的早期阶段所产生的“软件神话”,这些神话误导人们形成了一些错误概念和做法,严重地阻碍了计算机软件的开发,而且用错误方法开发出来的许多大型软件几乎根本无法维护,只好提前报废,造成大量人力、物力的浪费。最常见的软件神话包括以下几种。

- 只要拥有讲述如何开发软件的书籍,而且其中充满了标准与示例,就可以解决软件开发中的任何问题。
- 如果开发进度滞后,可以通过增加程序员来解决。
- 既然需求分析是非常复杂而且困难的,那就先开始做软件,反正软件是“软”的,可以随时改变。
- 一个项目的成功应该体现在所提交的程序,只要程序运行正常,项目也就结束了。创建软件工程中so要求的大量文档只会延缓开发进度。
- 在程序真正运行之前是没有办法评估其质量的。

随着计算机技术与软件开发技术的发展,许多人已经意识到这些神话是错误的。为了克服“软件神话”所带来的“软件危机”,人们进行了不断的探索。有人从制造机器和建筑楼房的过程中得到启示,无论是制造机器还是建造楼房都必须按照规划—设计—评审—施工(制造)—验

收—交付的过程来进行,那么在软件开发中是否也可以像制造机器与建造楼房那样,有计划、有步骤、有规范地开展软件的开发工作呢?答案是肯定的。于是,20 世纪 60 年代末用工程学的基本原理和方法来组织和管理软件开发全过程的一门新兴工程学科诞生了,这就是计算机软件工程学,通常简称为软件工程。

2. 软件工程的定义及其研究内容

那么,什么是软件工程呢?

自从 1968 年第一次提出软件工程的定义以来,软件工程的定义一直在不断地完善着。IEEE(IEEE 93)对软件工程的定义如下:软件工程是将系统化的、严格约束的、可量化的方法应用于软件的开发、运行和维护,即将工程化应用于软件。

通俗地说,软件工程是指导软件开发和维护的一门工程学科。它采用工程的概念、原理、技术和方法,把经过时间检验而证明是正确的管理技术和当前能够得到的最好的技术方法结合起来,用于开发和维护软件。

软件工程是一门综合性的交叉学科,它涉及哲学、计算机科学、工程科学、管理科学、数学及应用领域知识。软件工程研究的内容主要集中在软件的开发技术与管理两大方面。开发技术包括软件的开发模型、开发过程、开发方法、工具与环境等;管理技术包括人员组织、项目计划、标准与配置、成本估算、质量评价等。

从另一方面来说,软件工程又是一种层次化的技术(如图 1-3 所示)。因为任何工程方法都必须以质量控制为基础,因此质量控制是整个软件工程的基础。保证软件开发质量的前提条件是对软件工程中的各个过程进行有效的管理,为此必须为软件过程规定一系列的关键过程域,以此作为软件项目管理控制的基础,通过人员组织管理、项目计划管理、质量管理等环节来保证软件开发按时、按质量完成。软件工程中的“方法”提供了实现软件过程的技术,它涉及一系列的任务,包括需求分析、开发模型、设计、编码、测试和支持等。利用“工具”可以对软件过程与方法提供自动的或半自动的支持,在适当的软件工具辅助下,开发人员可以既快又好地做好软件开发工作,这些工具被称为 CASE(计算机辅助软件工程)工具。所以,一般将“过程”、“方法”和“工具”称为软件工程的三要素,这也是现代软件工程的研究内容。

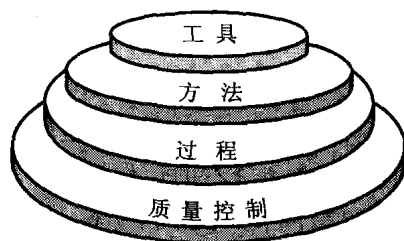


图 1-3 软件工程的层次

3. 软件工程的作用

软件工程的目标是提高软件的质量与生产率,最终实现软件的工程化管理、工业化生产。而质量与生产率往往是一对矛盾,软件的供需双方由于其利益的不同,关心的焦点也不同。质量是软件需方最关心的问题,它要求供方提供货真价实、满足需求的软件产品;而生产率则是供方最为关心的问题,它追求的是高的生产率,以获得最大的利益。因此如何在提高生产率的情况下开发出高质量的软件,就必然成为软件工程的主要目标,好的软件工程方法可以同时提高质量和生产率。

由于软件工程一开始是为了应对“软件危机”而提出的,如果在软件开发过程中能较好地利用软件工程的原理对软件开发的过程进行有效的管理,就可以充分保证软件开发的质量和生产率,反之就有可能造成项目的失败。下面就是正反两个方面的实例。

【成功案例】 美国联邦速递公司(FedEx)的管理信息系统

美国联邦速递公司是一个拥有数十亿美元资产、经营速递业务的大型企业。它拥有 643 架飞机、43 000 辆汽车、138 000 名员工,每天运送超过 310 万个包裹,通达全世界近 200 个国家和地区。该公司为适应管理的需要开发了覆盖整个公司的管理信息系统,从系统的架构、分析、开发直至运行、维护始终遵循需求至上的原则,将先进的软件工程的原理与方法贯穿整个软件开发过程,最终该项目取得了圆满的成功。通过管理信息系统,公司在任何时间都可以知道每件包裹在什么位置以及以后的运送路线,客户只要登录该公司的网站也可以得到同样的信息。后来,该系统又逐步扩展集成了从一个工厂的成品到送达用户之间的所有涉及分拣、运输、仓储、递送过程中每一步的状态数据。

【失败案例】 英国伦敦的急救服务管理信息系统

伦敦急救服务中心覆盖了 680 多万人口,每天接送 5 000 个病人,接听 2 500 个电话。为提高对急救电话的响应速度,更有效、及时地处理紧急情况,该中心开发了相应的急救信息管理系统,试图通过该系统对急救信息进行实时管理,最终目标是平均每 14 分钟响应一个电话,1992 年 10 月新系统正式投入运行。由于新的系统既没有经过严格的调试也没有经过完全的测试,尤其是满负荷下的测试,另一方面全体职员更没有经过对新系统的使用培训,使得系统在运行过程中发生了一系列致命的问题:有些紧急电话要花 30 分钟才能打进去,由于救护车延迟了 3 个小时,造成数十人死亡……伦敦急救服务中心的一位发言人说:“真是一场可怕的噩梦!”

以上正反两个方面的例子充分说明了软件工程在软件开发中的重要作用。

一般来说,软件企业的专业人员应该由以下几个层次构成:

① 高层管理人员。他们是软件企业的管理者,应具备软件专业的宏观知识、软件工程的理论知识、商业与资本的运作知识。他们是使用软件工程原理对企业进行管理的决策者。

② 项目经理与程序经理。他们是程序员的管理者,也是软件工程的拥有者与实践者。他们应具备系统分析与系统设计的能力以及项目管理方面的知识。他们必须使用软件工程的理论与方法来对软件的开发过程实施管理。在整个团队中,这种人的技术水平、办事效率应该是最高的,而且较高的人格魅力。

③ 程序员。他们是软件开发团队中的基础人员,应占整个企业员工的大多数,应具备阅读相关文档的技能、出色的程序设计与程序测试能力。他们要使用软件工程的理论与方法来实现软件项目的功能、接口与界面。

④ 营销人员。他们是软件企业的形象代表,必须通过他们才能将软件产品推销到使用单位。他们必须具备商业营销的知识、软件工程的基本知识,既要是某个行业领域的产品专家,又要成为该产品的实现顾问。

⑤ 售后服务人员。他们代表企业直接与用户打交道,实施软件的安装、运行与维护。他们也需要用软件工程的方法来对软件进行实施维护。

在以上几类人员中,前三类人员必须掌握软件工程的原理,对后两类人员只需要了解软件工程的相关知识并将之应用于实践之中。所以只要是在软件行业里工作,就必须重视软件工程、学好软件工程、用好软件工程,不断地将自己的实践经验上升到软件工程的理论与方法,又不断地用软件工程的理论与方法指导自己的实践,使自己得到升华与发展,这就是软件工程的作用。

实际上软件工程的作用是多方面的:对一个软件项目团队来说,实施软件工程可以保证在规