



普通高等教育“十一五”国家级规划教材

软件工程——理论与实践

(第2版)

Software Engineering: Theory and Practice
(2nd Edition)

许家玲 白忠建 吴磊 编著



高等教育出版社
Higher Education Press



普通高等教育“十一五”国家级规划教材

软件工程——理论与实践

(第2版)

许家珩 白忠建 吴磊 编著

高等教育出版社

内容提要

本书是普通高等教育“十一五”国家级规划教材,第1版是教育部新世纪网络课程建设工程“软件工程网络课程”子课题的配套教材。新版在继承第1版的“内容先进、注重实践、案例导向”特点的基础上,根据软件工程的最新发展进行了较大幅度的调整。教材内容涵盖 ACM/IEEE 制定的“软件工程教育知识体系”和教育部计算机专业教学指导委员会制定的《计算机专业规范》中对软件工程课程的要求,提供了更加丰富的案例,力求保证内容的科学性和先进性,并突出软件工程的实践性。

全书共13章。内容包括软件工程概述、软件需求工程、软件设计、面向对象方法基础、统一建模语言 UML、软件实现、软件复用与构件技术、软件测试、软件维护、软件工具与集成化环境、软件项目管理、软件能力成熟度模型 CMM。

随书附赠光盘一张,包含教材配套的电子教案和教材中部分习题的参考答案。

本书可作为高等院校计算机及相关专业软件工程课程的教材或教学参考书,也可供广大工程技术人员参考使用。

图书在版编目(CIP)数据

软件工程:理论与实践/许家玲,白忠建,吴磊编著.

2版.—北京:高等教育出版社,2009.10

ISBN 978-7-04-028044-9

I. 软… II. ①许…②白…③吴… III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2009)第 162311 号

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 100120
总 机 010-58581000
经 销 蓝色畅想图书发行有限公司
印 刷 潮河印业有限公司

开 本 787×1092 1/16
印 张 26
字 数 580 000

购书热线 010-58581118
咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2004年6月第1版
2009年10月第2版
印 次 2009年10月第1次印刷
定 价 36.00元(含光盘)

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

物料号 28044-00

前 言

软件工程作为一门指导计算机软件系统开发和维护的工程学科,近年对软件产业的支撑作用凸现,软件工程的应用水平和普及程度关系到我国的国际竞争力。

随着 Internet 的日益普及和广泛应用,全球软件产业呈现出网络化、服务化和全球化的发展趋势。软件产业将成为全球高科技产业发展最主要的推动力,这对我国软件产业的发展和国际化软件人才的培养提出了新的挑战。

教育部在关于“质量工程”的相关文件中明确指出学生的实践能力和创新精神亟待加强。因此,理论与实践密切结合,强化实践教学,改革实践教学内容,完善实践教学体系,推进教学方法的改革,加强优质资源建设,也成为我们编写教材第 2 版的指导思想。

本版在继承第 1 版的“内容先进、注重实践、案例导向”特点的基础上,根据软件工程的最新发展及作者二十多年从事软件工程课程教学和软件开发的实践经验,对全书的结构和内容进行了较大的调整,内容涵盖 ACM/IEEE 联合工作组制定的“软件工程教育知识体系”和教育部计算机专业教学指导委员会制定的《计算机专业规范》中对软件工程课程的要求,提供了更加丰富的案例,力求保证内容的科学性和先进性,突出软件工程的实践性,在内容的广度、深度和创新上都力争超过第 1 版。

全书共 13 章。第 1 章系统介绍软件工程的基本概念;第 2、3 章讨论需求和设计这两个软件开发重要阶段的任务和原则,并对需求获取、需求分析建模、软件体系结构的常用模式、用户界面设计等关键技术进行详细的讨论;第 4、5 章重点介绍应用 UML 建模的面向对象分析、设计及 RUP 统一过程;第 6~9 章讨论软件实现、测试和维护技术;第 10 章详细介绍软件开发环境;第 11、12 章详细介绍软件项目管理、CMM、ERP 等先进管理技术;第 13 章的软件工程课程设计是一个综合性的设计型实验,旨在培养学生的实践能力及创新能力,并给出 4 个采用 UML 面向对象建模的软件开发实例。

随书所附光盘包括两部分内容:第一部分是电子教案,第二部分是教材中部分习题的参考答案。

许家珩教授编写了第 1、2、4、5、8 章,并负责全书统稿,白忠建博士编写了第 3、6、7、11 章,吴磊博士编写了第 9、10、12、13 章。

复旦大学钱乐秋教授认真审阅了全书,并提出了十分宝贵的修改意见,在此表示诚挚的感谢。

在本书编写过程中,得到了许多教师和学生的鼓励和支持,黄迪明教授、曾翎副教授、彭德中

博士对本书的编写提出了许多有益的意见和建议,华为公司的吴知先生和北大青岛的侯晶女士为本书提供了部分软件开发的实例,黄金艳硕士为本书绘制和修改了部分插图。此外,高等教育出版社的有关人员也对本书从前期策划到编辑加工出版提供了必要的支持。在此对所有支持和帮助本书编写的人们表示衷心的感谢。

由于作者水平有限,诚恳希望广大读者批评指正。作者的 E-mail 为 jiayixu@uestc.edu.cn。

本课程是四川省精品课程,教学网站为 <http://125.71.228.229/wlxt/ncourse/se/web/soft/default.aspx>,欢迎读者浏览。

作者于电子科技大学
2009年7月

第1版前言

“软件工程”是一门指导计算机软件系统开发和维护的工程学科,是一门新兴的边缘学科。

自1968年在著名的NATO会议上为解决“软件危机”而提出“软件工程”这个概念以来,在不到40年的时间里,软件工程在理论和实践两个方面都取得了长足的进步,取得了大量研究成果。软件工程的应用水平已成为促进软件产业健康发展的关键。

软件是信息化的核心,软件产业是增长最快的朝阳产业,是高投入、高产出、无污染、低能耗的绿色产业。软件产业关系到国家经济发展和文化安全,体现了国家的综合实力,是决定21世纪国际竞争地位的战略产业。

随着计算机的日益普及和广泛应用,软件系统的规模和复杂度与日俱增,软件技术面临着新的挑战。大型复杂软件的开发是一项特殊的工程,不仅与传统工程一样,需要按照工程化的方法去组织管理软件的开发,而且软件开发更具有特殊性和复杂性。

我国在加入WTO以来,各行各业加速了信息化建设的步伐,也促进了软件工程的飞速发展。为了不断提高软件开发的质量和水平,必须学习、掌握和应用软件工程的基本理论和技术,了解软件经济学,提高软件产业的管理水平,才能使我国的软件产业在国际竞争中占领一席之地。

软件工程是一门实践性很强的课程,只有通过软件开发的实践才能真正掌握和应用软件工程的理论知识。特别注重实践性也是本书的一大特色。全书共分11章,系统介绍了软件工程的基本内容、开发及管理技术,同时重点介绍了软件工程的最新发展技术,如面向对象的方法(OO)、统一建模语言(UML)、软件成熟度模型(CMM)等。在第11章“软件工程课程设计”中,提供了一个综合性的设计型实验——“软件工程课程设计”,并提供了3个软件开发的典型案例。读者在学习软件工程课程的同时,选择完成一个适当的项目或课题,以软件工程课程设计的实践来促进软件工程理论的学习。本教材还介绍了面向对象的软件开发工具Rational Rose。作者开发的“软件工程网络课程”(已由高等教育出版社出版)提供了软件开发过程的指导和丰富的教学资源。

所附光盘包括两部分内容:第一部分是电子教案,第二部分是教材中所有习题的习题解答。

软件工程是一门处于前沿地位的重要学科,在迅速不断地发展。我们希望读者通过本书的学习,能将理论知识应用于软件开发的实践中,并在实践中不断创新。

本书由黄迪明教授主审,许家珩教授主编。第1~5章、第11章由许家珩教授编写,第8章、第10章由曾翎博士编写;第6章、第9章由彭德中博士编写,第7章由侯晶硕士编写。同时还要

感谢白忠建老师、刘智、杨臣、陈昌波及吴知硕士为本书提供了大量宝贵的素材,程永新、陈科硕士对本书进行了仔细的校对。

由于作者水平有限,书中疏漏、欠妥、谬误之处在所难免,恳请读者批评指正。

作者于电子科技大学
2004年5月

参 考 网 站

- [1] <http://www.comp.mq.edu.au/books/maciaszek>
- [2] <http://www.booksitesnet/maciaszek>
- [3] <http://www.uml.org.cn/RequirementProject/xqgc.asp>
- [4] <http://www.uml.org.cn/appCase/appCase.asp>
- [5] <http://www.sei.cmu.edu>
- [6] <http://www.rational.com>

目 录

第1章 软件工程概述	1	2.1.2 功能需求与非功能需求	20
1.1 软件工程的产生和发展	1	2.2 需求工程过程	23
1.1.1 软件工程的发展过程	1	2.2.1 需求工程的基本活动	23
1.1.2 软件危机	2	2.2.2 需求的获取和分析	24
1.1.3 软件工程的定义	3	2.2.3 需求的有效性验证	25
1.1.4 软件工程研究的内容	3	2.2.4 需求管理	25
1.1.5 软件工程的基本原则	5	2.3 软件需求获取技术	26
1.2 软件与软件过程	5	2.3.1 需求获取的困难	27
1.2.1 软件的概念和特点	5	2.3.2 需求获取方法	27
1.2.2 软件工程过程	7	2.4 需求分析与建模	29
1.3 软件过程模型	8	2.4.1 需求分析方法	29
1.3.1 瀑布模型	8	2.4.2 结构化分析方法	30
1.3.2 增量模型	9	2.4.3 面向对象的分析方法	36
1.3.3 螺旋模型	9	2.5 软件需求案例分析	37
1.3.4 喷泉模型	9	2.5.1 案例1——医院病房监护系统	37
1.3.5 智能模型	11	2.5.2 案例2——网上竞价拍卖系统	41
1.4 软件开发方法	12	小结	43
1.4.1 结构化开发方法	12	习题二	43
1.4.2 原型化开发方法	12	第3章 软件设计	46
1.4.3 面向对象的开发方法	13	3.1 软件设计阶段的任务和目标	46
1.5 软件工具与软件开发环境	14	3.1.1 软件设计在开发阶段中的 重要性	46
1.5.1 CASE发展的两个阶段	14	3.1.2 软件设计阶段的任务	47
1.5.2 软件开发环境的分类	16	3.2 软件体系结构设计	48
小结	16	3.2.1 仓库模型	49
习题一	17	3.2.2 分布式结构	50
第2章 软件需求工程	19	3.2.3 多处理器结构	51
2.1 软件需求的基本概念	19	3.2.4 客户-服务器模型	52
2.1.1 软件需求的任务	19		

3.2.5 多层应用模型	54	4.4 面向对象的设计	91
3.2.6 分布式对象结构	55	4.4.1 面向对象设计的准则	91
3.2.7 中间件	57	4.4.2 面向对象设计的基本任务	92
3.2.8 其他体系结构	59	4.5 典型的面向对象方法	93
3.3 模块分解	60	4.5.1 Booch 方法	94
3.3.1 模块分解的目的	60	4.5.2 Coad/Yourdon 方法	96
3.3.2 模块间的关系	61	4.5.3 对象模型技术	99
3.3.3 模块的独立性	62	4.5.4 OOSE 方法	104
3.3.4 信息隐蔽	65	小结	106
3.4 详细设计描述工具	65	习题四	106
3.4.1 程序流程图	66		
3.4.2 N-S 图	66	第 5 章 统一建模语言 UML	109
3.4.3 PAD 图	66	5.1 UML 概述	109
3.5 用户界面设计	67	5.1.1 UML 的形成	109
3.5.1 用户界面设计的特性与设计任务	68	5.1.2 UML 的特点	110
3.5.2 用户界面设计的基本原则	69	5.1.3 UML 建模及其构成	111
3.5.3 用户界面的基本类型	69	5.1.4 UML 的图形表示	112
3.5.4 信息输入输出界面	71	5.1.5 通用模型元素	113
3.5.5 MVC 模式	78	5.2 建立用例模型	115
小结	80	5.2.1 需求分析与用例建模	115
习题三	80	5.2.2 确定执行者	116
		5.2.3 确定用例	119
第 4 章 面向对象方法基础	83	5.2.4 确定用例之间的关系	121
4.1 面向对象方法概述	83	5.2.5 用例建模实例	122
4.1.1 什么是面向对象方法	83	5.3 建立静态模型	126
4.1.2 面向对象方法的主要特点	84	5.3.1 类与类图	126
4.2 面向对象的基本概念	85	5.3.2 确定类之间的关系	131
4.2.1 对象与类	85	5.3.3 类图应用举例	137
4.2.2 继承	86	5.3.4 包图	139
4.2.3 多态性	87	5.4 建立动态模型	142
4.2.4 消息	87	5.4.1 消息	142
4.3 面向对象的分析	88	5.4.2 状态图	142
4.3.1 需求分析中的问题	88	5.4.3 顺序图	145
4.3.2 OOA 的特点	89	5.4.4 协作图	148
4.3.3 OOA 的基本任务和分析过程	90	5.4.5 活动图	151
		5.4.6 动态建模应用举例	153

5.5 建立实现模型	155	第7章 软件复用与构件技术	193
5.5.1 构件图	155	7.1 软件复用概述	193
5.5.2 配置图	157	7.1.1 软件复用的基本概念	193
5.6 UML 2.0	159	7.1.2 软件复用的级别	194
5.6.1 UML 2.0 简介	159	7.1.3 软件复用的形式	195
5.6.2 UML 2.0 的可视化图	161	7.2 软件复用的实施与过程	196
5.6.3 UML 2.0 的顺序图	163	7.2.1 软件复用技术的实施	196
5.7 RUP 统一过程及其应用	165	7.2.2 基于复用的软件开发过程	197
5.7.1 UML 与 RUP 统一过程	165	7.3 可复用构件与构件工程	198
5.7.2 RUP 的特点	166	7.3.1 可复用构件	198
5.7.3 RUP 的二维开发模型	167	7.3.2 基于构件的软件工程	199
5.7.4 RUP 的迭代开发模式	170	7.4 领域工程分析和基于构件	
小结	171	的开发	199
习题五	171	7.4.1 领域分析	199
第6章 软件实现	174	7.4.2 构件的开发与构件库	200
6.1 程序设计语言的选择	174	7.4.3 基于构件的开发	202
6.1.1 程序设计语言的分类	174	7.5 基于构件的软件开发特点	203
6.1.2 程序设计语言的基本组成	175	7.5.1 软件开发的质量	203
6.1.3 程序设计语言选择准则	176	7.5.2 软件开发的效率	203
6.2 结构化程序设计	181	7.5.3 软件开发的成本	204
6.2.1 结构化程序设计概念和特点	181	7.6 软件构件技术的技术规范	204
6.2.2 结构化程序设计的基本原理	182	7.6.1 OMG 的 CORBA	204
6.2.3 优化设计	182	7.6.2 微软的 COM	206
6.3 程序设计风格	183	7.6.3 Sun 的 EJB	208
6.3.1 源程序文件	183	7.6.4 Web 服务	210
6.3.2 语句构造方法	185	小结	212
6.3.3 数据说明方法	185	习题七	213
6.3.4 输入/输出技术	186	第8章 软件测试	214
6.4 算法与程序效率	186	8.1 软件测试的基本概念	214
6.4.1 算法转换过程中的指导原则	186	8.1.1 软件测试的重要性和目的	214
6.4.2 影响效率的因素	188	8.1.2 软件测试的特点和基本原则	216
6.5 软件代码审查	190	8.1.3 软件测试过程	219
小结	191	8.1.4 静态分析与动态测试	222
习题六	191	8.2 白盒法测试	224
		8.2.1 逻辑覆盖	224

8.2.2	语句覆盖	224	9.1.4	软件维护的代价	261
8.2.3	判定覆盖	225	9.2	软件维护的过程	262
8.2.4	条件覆盖	226	9.2.1	维护组织	262
8.2.5	判定-条件覆盖	227	9.2.2	维护工作的流程	263
8.2.6	条件组合覆盖	227	9.2.3	维护工作的组织管理	263
8.3	黑盒法测试	229	9.3	软件维护技术	264
8.3.1	等价分类法	229	9.3.1	面向维护的技术	264
8.3.2	边界值分析法	230	9.3.2	维护支援技术	265
8.3.3	错误推测法	231	9.3.3	维护档案记录	265
8.3.4	因果图法	231	9.3.4	维护工作评价	265
8.4	软件测试的策略	234	9.4	软件可维护性	266
8.4.1	单元测试	234	9.4.1	软件可维护性的定义	266
8.4.2	集成测试	237	9.4.2	提高可维护性的方法	268
8.4.3	确认测试	240	9.5	逆向工程和再工程	272
8.4.4	系统测试	241	9.5.1	逆向工程	272
8.4.5	α 测试和 β 测试	242	9.5.2	软件重构	273
8.4.6	验收测试	242	9.5.3	再工程的成本/效益分析	273
8.4.7	综合测试策略	243	9.5.4	再工程的风险分析	274
8.5	软件调试	243	小结	274	
8.5.1	软件调试过程	243	习题九	274	
8.5.2	软件调试策略	244			
8.6	面向对象的测试	246	第 10 章	软件工具与集成化环境	276
8.6.1	面向对象测试的特点	247	10.1	软件工具	276
8.6.2	面向对象测试的类型	247	10.1.1	软件开发工具	276
8.6.3	分析模型测试	250	10.1.2	软件维护工具	278
8.6.4	类的测试	253	10.1.3	软件管理与支持工具	280
8.6.5	类模型测试	254	10.2	集成化 CASE 环境	281
8.6.6	面向对象的测试用例	255	10.2.1	概述	281
小结	255	10.2.2	集成化的 CASE 开发环境的 体系结构	283	
习题八	256	10.3	软件开发工具 Rational Rose	286	
第 9 章	软件维护	258	10.3.1	Rose 工具简介	286
9.1	软件维护的基本概念	258	10.3.2	业务用例图	287
9.1.1	软件维护的目的	258	10.3.3	用例图	288
9.1.2	软件维护的类型	258	10.3.4	类图	290
9.1.3	软件维护的特性	260	10.3.5	协作图与顺序图	292

10.3.6 活动图	294	11.5.3 软件项目的跟踪与控制	322
10.3.7 状态图	295	11.6 企业资源规划 ERP	323
10.3.8 构件图和部署图	297	11.6.1 资源管理发展过程	323
10.4 软件开发工具 Power-Designer	298	11.6.2 ERP 系统的管理思想	324
10.4.1 PowerDesigner 工具简介	298	11.6.3 应用 ERP 与企业的关系	325
10.4.2 PowerDesigner 的 6 个模块	299	11.6.4 ERP 的风险及其预防	326
10.4.3 PowerDesigner 与 Rose 建模工具 的对比	300	11.6.5 ERP 应用成功的标志	326
小结	301	11.6.6 SAP ERP 简介	327
习题十	301	小结	329
习题十一	301	习题十一	329
第 11 章 软件项目管理	303	第 12 章 软件能力成熟度模型 CMM	331
11.1 软件项目管理概述	303	12.1 CMM 概述	331
11.1.1 软件项目管理的特点	303	12.1.1 CMM 发展历史与主要用途	331
11.1.2 软件项目管理的主要活动	304	12.1.2 软件过程成熟度的基本概念	332
11.1.3 软件项目风险管理	305	12.1.3 软件过程的成熟度等级	333
11.2 软件项目可行性研究	308	12.2 CMM 的内部结构	335
11.2.1 可行性研究的任务	309	12.2.1 关键过程域	335
11.2.2 系统的描述	309	12.2.2 关键实践	337
11.2.3 可行性研究报告	310	12.2.3 共同特性	337
11.3 软件项目成本估算技术	311	12.3 CMM 的应用	337
11.3.1 影响成本估算的因素	311	12.3.1 软件过程评估与软件能力 评价	337
11.3.2 成本估算模型	312	12.3.2 软件过程改进	339
11.3.3 专家估算模型	313	12.4 CMM 的实施与评估	339
11.3.4 IBM 估算模型	313	12.4.1 软件过程评估的必要性	340
11.3.5 Putnam 估算模型	314	12.4.2 软件过程评估参考模型	341
11.3.6 COCOMO 模型	314	12.4.3 CMM 评估的执行步骤	343
11.3.7 成本/效益分析	316	12.4.4 软件企业如何实施 CMM	346
11.4 软件项目组织与人员管理	317	12.4.5 CMM 与 ISO 9000 标准	349
11.4.1 软件开发进度计划	317	12.5 软件能力成熟度模型集成 CMMI	350
11.4.2 人员配备与组织	319	12.5.1 CMMI 的产生与发展	350
11.4.3 软件开发小组与软件 生产率	320	12.5.2 CMMI 的基本思想	351
11.5 软件质量保证	321	12.5.3 CMMI 的基本内容	352
11.5.1 软件质量因素的定义	321		
11.5.2 软件质量保证工作	322		

小结	354	13.2.4 编码与测试	358
习题十二	354	13.2.5 验收测试	358
第 13 章 软件工程课程设计	356	13.3 案例分析	359
13.1 课程设计目的和要求	356	13.3.1 案例一:ATM 系统	359
13.1.1 目的和要求	356	13.3.2 案例二:网上拍卖系统	366
13.1.2 命题原则	356	13.3.3 案例三:会议管理系统	373
13.2 课程设计步骤安排	357	13.3.4 案例四:仓库信息管理系统	388
13.2.1 确定课题	357	参考文献	398
13.2.2 需求分析	357	参考网站	400
13.2.3 软件设计	357		

第1章 软件工程概述

1.1 软件工程的产生和发展

软件是信息化的核心,国民经济、国防建设、社会发展及人民生活都离不开软件。软件产业关系到国家经济发展和文化安全,体现了国家综合实力,是决定21世纪国际竞争地位的战略产业。

软件工程是一门指导计算机软件系统开发和维护的工程学科,是应用计算机科学与技术以及工程管理的原则与方法,实现满足用户要求的软件产品定义、开发和维护的过程。

软件工程是一门新兴的边缘学科,它涉及计算机科学、工程科学、管理科学、数学等多门学科,其研究范围很广——不仅包括软件系统的开发方法和技术、管理技术,还包括软件工具、环境及软件开发的规范。

随着我国信息化建设的深入发展,大力推广和应用软件工程的开发及管理技术,提高软件工程的应用水平,对促进我国软件产业与国际接轨,推动软件产业的迅速发展具有十分重要的意义。

1.1.1 软件工程的发展过程

软件工程(software engineering, SE)是在克服20世纪60年代末所出现的“软件危机”的过程中逐渐形成与发展的。在40年的时间里,软件工程在理论和实践两方面都取得了长足的进步。它的发展已经历了4个重要阶段。

1. 第一代软件工程

20世纪60年代末,软件生产主要采用“生产作坊”方式。随着软件需求量、规模及复杂度的迅速增大,生产作坊的方式已不能够适应软件生产的需要,出现了所谓的“软件危机”,其主要表现为软件生产效率低下、软件产品质量低劣,大量质量低劣的软件涌入市场后不久或在开发过程中就夭折。由于“软件危机”的不断扩大,国际软件界面临着巨大的灾难,软件产业濒临崩溃。

为了克服“软件危机”,1968年在北大西洋公约组织(NATO)举行的软件可靠性学术会议上第一次提出了“软件工程”的概念,其核心是将软件开发纳入工程化的轨道,以保证软件开发的效率和质量。

此后,逐渐形成了软件工程的基本概念、框架、技术和方法,以结构化开发方法、Jackson方法

等为代表的软件开发方法也成为这一阶段的主要开发方法。这一阶段又称为传统的软件工程阶段。

2. 第二代软件工程

从20世纪80年代中期开始,以Smalltalk为代表的面向对象的程序设计语言相继推出,面向对象的方法与技术得到迅速发展;从20世纪90年代起,软件工程研究的重点从程序设计语言逐渐转移到面向对象的分析与设计,演化成为一种完整的软件开发方法和系统的技术体系。

20世纪90年代以来,形成了Booch方法、OOSE、OMT等许多面向对象开发方法的流派,面向对象的方法逐渐成为软件开发的主流。尤其是1997年1月,综合了各种面向对象方法优点的统一建模语言UML 1.0版的正式推出,使面向对象的方法得到进一步发展,所以这一阶段又称为对象工程。

3. 第三代软件工程

随着软件规模和复杂度的不断增大,开发人员也随之增多,开发周期相应延长,加之软件是知识密集型的逻辑思维产品,这些都增加了软件工程管理的难度。人们在软件开发的实践过程中逐渐认识到:提高软件生产率、保证软件质量的关键是对“软件过程”的控制和管理,是软件开发和维护中的管理和支持能力。因此提出了对软件项目管理的计划、组织、成本估算、质量保证、软件配置管理等技术与策略,逐步形成了软件过程工程。

4. 第四代软件工程

20世纪90年代起,软件复用和基于构件(component)的开发方法取得重要进展,软件系统的开发可通过使用现成的可复用构件组装完成,而无须从头开始构造,以此达到提高效率和质量、降低成本的目的。软件复用技术及构件技术的发展,为克服软件危机提供了一条有效途径,已成为当前软件工程的重要研究方向,这一阶段又称为构件工程。

1.1.2 软件危机

1. 软件危机的产生

软件开发过程是一种高密集度的脑力劳动,需要投入大量的人力、物力和财力。20世纪60年代末,软件的规模越来越大,复杂度不断增加,软件的需求量也不断增大,而当时生产作坊式的软件开发模式及技术已远远不能满足软件发展的需要,致使大量质量低劣的软件产品涌向市场。一些大型软件系统的开发,不是费用大大超过了预算,就是系统比原计划推迟了好多年。有的系统不符合用户的需求而被废弃,有的系统则无法进行修改维护。这些都使软件生产面临巨大困难和严重危机,即“软件危机”(software crisis)。典型的例子如下。

IBM公司的OS/360系统,共约100万条指令,花费了5000个人年,经费达数亿美元,而结果却令人沮丧:错误多达2000个以上,系统根本无法正常运行。OS/360系统的负责人Brooks这样描述开发过程的困难和混乱:“像巨兽在泥潭中作垂死挣扎,挣扎得越猛,泥浆就沾得越多,最后没有一个野兽能够逃脱淹没在泥潭中的命运……”。

1967年苏联的“联盟一号”载人宇宙飞船,由于软件中一个小数点的疏忽,导致返航时打不

开降落伞,当进入大气层时因摩擦力太大而烧毁,造成机毁人亡的巨大损失。

被称为 20 世纪世界上最精心设计、花费了巨额投资的美国阿波罗登月飞行计划的软件,也仍然没能避免出错。例如,阿波罗 8 号由于太空飞船的一个计算机软件错误,造成存储器的一部分信息丢失;阿波罗 14 号在飞行的 10 天中出现了 18 个软件错误。

软件危机已严重威胁了世界软件产业的发展。

2. 软件危机的表现

20 世纪 60 年代末期所发生的软件危机,反映在软件可靠性没有保障、软件维护工作量大、费用不断上升、进度无法预测、成本增长无法控制、程序人员无限度地增加等各个方面,以致形成人们难以控制软件开发的局面。

软件危机主要表现在两个方面:

- ① 软件产品质量低劣,甚至在开发过程中就夭折。
- ② 软件生产率,不能满足需要。

软件危机所造成的严重后果已致使世界各国的软件产业危机四伏,面临崩溃,克服软件危机刻不容缓。从 NATO 会议以来,世界各国的软件工作者为克服软件危机进行了许多开创性的工作,在软件工程的理论研究和工程实践两个方面都取得长足的进步,缓解了软件危机。但距离彻底地克服软件危机这个软件工程的最终目标还任重道远,需要软件工作者付出长期艰苦的努力。

1.1.3 软件工程的定义

虽然对软件工程有着众多的定义,但是它们的基本思想都是强调在软件开发过程中应用工程化原则的重要性。

例如,1983 年,IEEE(电气和电子工程师协会)所下的定义是:软件工程是开发、运行、维护和修复软件的系统方法。1990 年,IEEE 又将定义更改为:对软件开发、运作、维护的系统化的、有规范的、可量化的方法之应用,即是对软件的工程化应用。

2004 年由 IEEE/ACM 联合发布的 CCSE 2004 报告强调了对软件工程的新定义,即软件工程是“以系统的、学科的、量化的途径,把工程应用于软件的开发、运营和维护;同时,开展对上述过程中各种方法和途径的研究”。

从软件工程的定义可见,软件工程是一门指导软件系统开发的工程学科,它以计算机理论及其他相关学科的理论为指导,采用工程化的概念、原理、技术和方法进行软件的开发和维护,把经实践证明的科学的措施与最先进的技术方法结合起来,软件工程研究的目标是“以较少的投资获取高质量的软件”。

1.1.4 软件工程研究的内容

软件工程的研究和实践涉及人力、技术、资金、进度的综合管理,是开展最优化生产活动的过程;软件工程是一门新兴的边缘学科,涉及的学科多,研究的范围广。归结起来软件工程研究的主要内容有以下 4 个方面:方法与技术、工具与环境、管理技术、标准与规范。