

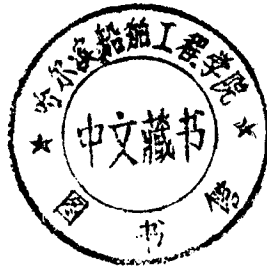
TP31
3150
(3)

412587

软件工程导论

(第三版)

张海藩 编著



清华大学出版社

(京)新登字 158 号

内 容 提 要

本书第二版和第一版曾先后荣获第三届和第二届全国普通高等学校工科电子类专业优秀教材一等奖、二等奖。为适应面向对象技术迅速崛起的情况,作者总结五年来教学和科研的经验,对原书内容作了调整和增删,变动超过 50%,成为第三版。书中全面系统地介绍了软件工程的观念、原理和典型的技术方法。本书的特点是既注重系统性和科学性,又注重实用性;既有原理性论述,又有丰富的实例与之配合,特别是正文后面的两个附录,分别讲述了用面向对象方法和结构化方法开发两个实际软件的过程,对读者深入理解软件工程学很有帮助,也是上机实习的好材料。本书正文共 13 章,第一章是概述,第二章至第八章顺序讲述软件生命周期各阶段的任务、过程、结构化方法和工具,第九章至第十二章分别讲述面向对象方法学引论、面向对象分析、面向对象设计和面向对象实现,第十三章介绍软件工程使用的管理技术。

本书可作为高等院校“软件工程”课程的教材或教学参考书,也可供有一定实际经验的软件工作人员和需要开发应用软件的广大计算机用户阅读参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

软件工程导论/张海藩编著. —3. —北京:清华大学出版社,1998

ISBN 7-302-02799-4

I. 软… II. 张… III. 软件工程-概论 IV. TP311

中国版本图书馆 CIP 数据核字(98)第 00041 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

因特网地址: www.tup.tsinghua.edu.cn

责任编辑:贾仲良

印刷者:北京昌平环球印刷厂

发行者:新华书店总店北京科技发行所

开本: 787×1092 1/16 印张: 23.25 字数: 552 千字

版次: 1998 年 1 月第 3 版 1998 年 7 月第 3 次印刷

书号: ISBN 7-302-02799-4/TP·1463

印数: 212501~223500

定价: 22.50 元

前 言

计算机工业发达国家在发展软件的过程中曾经走过不少弯路,受过许多挫折,至今仍然经受着“软件危机”的困扰。人们开发优质软件的能力大大落后于计算机硬件日新月异的进展和社会对计算机软件不断增长的需求,这种状况已经严重妨碍了计算技术的进步。

为了摆脱软件危机的困扰,一门研究软件开发与维护的普遍原理和技术的工程学科——软件工程学——从60年代末期开始迅速发展起来了,现在它已经成为计算机科学技术的一个重要分支,一个异常活跃的研究领域。严格遵循软件工程方法论可以大大提高软件开发的成功率,能够显著减少软件开发和维护中的问题。

软件工程学(通常简称软件工程)研究的范围非常广泛,包括技术方法、工具和管理等许多方面,软件工程又是一门迅速发展的新兴学科,新的技术方法和工具不断涌现,真可谓五花八门层出不穷。因此,在一本书中不可能包含软件工程的全部内容。本书《软件工程导论》是软件工程的入门介绍,着重从实用角度讲述软件工程的基本原理、概念和技术方法,同时也尽量注意书的全面性和系统性。希望本书既能对实际的软件开发工作有所帮助,又能为读者在今后深入研究这门学科奠定良好的基础。

本书正文共十章。第一章概括介绍软件工程学产生的历史背景以及它的基本原理、概念和方法。第二章到第九章按软件生存周期的顺序介绍各个阶段的任务、过程、方法和工具。第十章着重讨论软件工程的管理技术。正文后面有两个附录,比较完整地介绍了两个实际软件的开发。附录A着重讲述从问题定义到实现的过程,把这个具体例子和课文前几章的内容结合起来学习,有助于加深对一些基本概念和方法的理解。附录B讲述一个交互式正文编辑程序的设计和描述,它是上机实习的好材料(例如,可以实习把设计翻译成程序、设计测试方案以及维护的方法)。

《软件工程导论》可以为多种读者服务。本书通俗易懂,实例丰富,既有对多种常见方法的全面概括介绍,又有对一种典型方法的深入详尽介绍,很适合于有一定实践经验的软件工作者和广大计算机用户参考或自学;对于高等院校计算机系高年级本科生和研究生来说,本书可以做为软件工程课程的教材。

本书根据编者最近几年在北京大学校外多次讲授“软件工程概论”课所用的讲义改写而成,改写时充分考虑了在校内外讲授这门课程时广大学员的建议和要求,并且增加了最近收集到的大量新材料。

编者在美国马里蓝大学进修期间,听过该校朱耀汉(Yaohan Chu)教授讲授的软件工程课,并且在朱教授指导下从事过一些软件开发工作,这些都为本书的编写奠定了基础。本书编写前征求了复旦大学计算机系副教授潘锦平同志对写作大纲的意见,初稿写出后又征求了潘锦平和北大计算机系副系主任许卓群等同志对内容编排的意见,承蒙他们给

予真诚的鼓励并且提出了许多宝贵的具体建议。此外,本书编写过程中董士海副教授曾经提供了许多很有价值的材料。谨在此向他们表示衷心的感谢!

本书不当之处敬请广大读者不吝赐教。

编者

1986年

修订版前言

本书第一版自 1987 年 6 月出版以来,颇受读者欢迎,不少高校用其作为“软件工程概论”课的教材。然而第一版的内容主要取材于 1985 年以前出版的书籍或文章和著者的实践总结,经过五年多的时间,这一学科又有了不少新的进展,为了跟踪学科发展方向,更好地为广大读者服务,编者对原书作了认真修订。

这次修订的指导思想是,保持原书系统性较强,内容比较全面,有丰富的实例与原理性论述紧密配合的特点,增加了反映学科最新发展方向的新内容,同时又适当压缩全书的篇幅,以降低成本,减轻读者负担。为此在修订时对书的取材作了十分慎重的斟酌,删掉了前后重复的内容,去掉了对理解本书的内容不是十分必要的图表和文字,占篇幅比较多的原附录 A 也被割爱了;增加了近年来比较流行的原型法、面向对象的程序设计和软件再用技术的介绍,强调了软件工程与人工智能相互结合、互相促进的发展趋势。对附录 B 介绍的编辑程序的功能作了扩充,不仅增加了编辑命令,还增加了输入和编辑汉字的功能。

此次修订还适当加强了对日本软件工程方法的介绍,例如,第五章增加了对日立公司发明的 PAD 图的介绍,第八章讲述了日立预测法的内容。

另一个比较重要的改动,是使书中使用的术语与近年来国内的习惯用法保持一致,对其他文字也作了进一步的加工和润色。

为便于教学,此次修订在每章后面都附上了适当数量的习题。

编者

1990 年 11 月
于北京信息工程学院

第三版前言

本书修订版(即第二版)自1992年6月出版以来,受到更多读者的欢迎,在不足五年的时间内共印刷了13次,台湾于1993年2月出版了本书(修订版)的繁体字版。有关部门领导和同行专家也对本书给予了充分肯定和热情鼓励,于1995年12月把本书(修订版)评为第三届全国普通高等学校工科电子类专业优秀教材一等奖。

五年来计算机软件技术又有了突飞猛进的进步,特别是面向对象的软件技术在我国已经逐渐进入实用阶段,越来越多的软件工作者开始使用面向对象技术开发自己的应用系统。为了跟踪学科发展方向,更好地为广大读者服务,编者再次对原书作了认真修订,写出了第三版。

在第三版中,编者吸取国内外论述面向对象方法学众多书刊的精华,并结合自己使用面向对象方法从事科学研究、开发软件的经验 and 体会,对面向对象方法作了全面、系统、深入的介绍。第三版新增加了4章新内容,结合丰富的实例集中讲述面向对象方法学。此外,新编写的附录A,讲述了用面向对象方法分析、设计一个简化的C++类库管理系统的过程。读者通过这个实例,不仅可以对面向对象分析和面向对象设计有更具体、更深入的认识,还可以在这份材料的基础上进行用面向对象方法分析、设计和实现软件的实习。

在努力推广新技术的同时,编者认为对传统的结构分析和结构设计技术也应该“一分为二”。事实上,结构分析和结构设计技术现在仍然是人们在开发软件过程中使用得非常广泛的方法,在相当长一段时间内这种方法还会有生命力。特别是作为一本教材来说,更应该把基本概念、原理、技术和方法讲深讲透,因此,在第三版中保留了第二版的基本内容,并且根据编者多年从事软件工程教学和研究的体会,补充了一些必要的内容(例如,软件工程的基本原理,信息建模的概念和方法等)。对于第二版中已经是陈旧了的内容,则坚决删掉。近年来软件蓝图方法应用得并不广泛,因此在第三版中删掉了介绍这个方法的原第六章,并对附录B作了相应的修改,不再使用软件蓝图描述设计结果。

我的学生张劲松和张展新用Visual C++ 4.0成功地实现了附录A中所述的C++类库管理系统。谨在此向他们表示衷心的感谢!

编者

1997年6月

目 录

前言	I
修订版前言	Ⅲ
第三版前言	IV
第一章 软件危机与软件工程	1
1.1 软件危机	1
1.1.1 什么是软件危机	1
1.1.2 产生软件危机的原因	2
1.1.3 解决软件危机的途径	5
1.2 软件工程	5
1.2.1 软件工程的基本原理	5
1.2.2 软件工程的传统途径	7
1.2.3 生命周期各阶段的基本任务	8
1.2.4 瀑布模型	12
1.3 技术审查和管理复审	13
1.3.1 进行审查和复审的必要性	13
1.3.2 技术审查的标准和方法	14
1.4 小结	15
习题一	16
第二章 可行性研究	18
2.1 可行性研究的任务	18
2.2 可行性研究的步骤	18
2.2.1 复查系统规模和目标	19
2.2.2 研究目前正在使用的系统	19
2.2.3 导出新系统的高层逻辑模型	19
2.2.4 重新定义问题	20
2.2.5 导出和评价供选择的解法	20
2.2.6 推荐行动方针	20
2.2.7 草拟开发计划	21
2.2.8 书写文档提交审查	21
2.3 系统流程图	21
2.3.1 符号	21
2.3.2 例子	22
2.3.3 分层	23

2.4	数据流图	23
2.4.1	符号	23
2.4.2	例子	25
2.4.3	命名	27
2.4.4	用途	28
2.5	数据字典	29
2.5.1	数据字典的内容	30
2.5.2	定义数据的方法	30
2.5.3	数据字典的用途	31
2.5.4	数据字典的实现	31
2.6	成本/效益分析	32
2.6.1	成本估计	33
2.6.2	成本/效益分析的方法	34
2.7	小结	36
	习题二	36
第三章	需求分析	38
3.1	需求分析的任务	38
3.1.1	确定对系统的综合要求	38
3.1.2	分析系统的数据要求	39
3.1.3	导出系统的逻辑模型	39
3.1.4	修正系统开发计划	39
3.1.5	开发原型系统	39
3.2	分析过程	40
3.2.1	沿数据流图回溯	40
3.2.2	用户复查	41
3.2.3	细化数据流图	41
3.2.4	修正开发计划	42
3.2.5	书写文档	42
3.2.6	审查和复审	43
3.3	概念模型和规范化	43
3.3.1	ER 模型	43
3.3.2	范式	45
3.4	图形工具	45
3.4.1	层次方框图	45
3.4.2	Warnier 图	46
3.4.3	IPO 图	47
3.5	验证软件需求	48
3.5.1	从哪些方面验证软件需求的正确性	48

3.5.2	验证软件需求的方法	48
3.5.3	用于需求分析的软件工具	49
3.5.4	超高级语言和第四代语言	50
3.6	小结	51
	习题三	52
第四章	总体设计	54
4.1	总体设计的过程	54
4.1.1	设想供选择的方案	54
4.1.2	选取合理的方案	55
4.1.3	推荐最佳方案	55
4.1.4	功能分解	55
4.1.5	设计软件结构	55
4.1.6	数据库设计	56
4.1.7	制定测试计划	56
4.1.8	书写文档	56
4.1.9	审查和复审	57
4.2	软件设计的概念和原理	57
4.2.1	模块化	57
4.2.2	抽象	58
4.2.3	信息隐蔽和局部化	59
4.2.4	模块独立	59
4.3	启发式规则	62
4.3.1	改进软件结构提高模块独立性	62
4.3.2	模块规模应该适中	62
4.3.3	深度、宽度、扇出和扇入都应适当	62
4.3.4	模块的作用域应该在控制域之内	63
4.3.5	力争降低模块接口的复杂程度	63
4.3.6	设计单入口单出口的模块	64
4.3.7	模块功能应该可以预测	64
4.4	图形工具	64
4.4.1	层次图和 HIPO 图	64
4.4.2	结构图	65
4.5	面向数据流的设计方法	66
4.5.1	概念	67
4.5.2	变换分析	68
4.5.3	事务分析	74
4.5.4	设计优化	75
4.6	小结	75

习题四	76
第五章 详细设计	80
5.1 结构程序设计.....	80
5.2 详细设计的工具.....	83
5.2.1 程序流程图	83
5.2.2 盒图(N-S图)	84
5.2.3 PAD图	84
5.2.4 判定表	86
5.2.5 判定树	87
5.2.6 过程设计语言(PDL)	87
5.2.7 模块开发文件夹	88
5.3 Jackson 程序设计方法	88
5.3.1 Jackson 图	89
5.3.2 改进的 Jackson 图	90
5.3.3 Jackson 方法	90
5.4 Warnier 程序设计方法	95
5.4.1 Warnier 方法	95
5.4.2 Warnier 方法的辅助技术	100
5.5 程序复杂程度的定量度量	104
5.5.1 McCabe 方法	104
5.5.2 Halstead 方法.....	106
5.6 小结	107
习题五.....	107
第六章 编码.....	113
6.1 程序设计语言	113
6.1.1 程序设计语言分类.....	113
6.1.2 程序设计语言的特点.....	114
6.1.3 选择一种语言.....	117
6.2 程序设计途径	118
6.2.1 写程序的风格.....	119
6.2.2 程序设计方法论.....	121
6.2.3 程序设计自动化.....	121
6.2.4 程序设计工具.....	122
6.3 小结	124
习题六.....	124
第七章 测试.....	126
7.1 基本概念	126
7.1.1 软件测试的目标.....	127

7.1.2	黑盒测试和白盒测试	127
7.1.3	软件测试的步骤	128
7.1.4	测试阶段的信息流	129
7.2	单元测试	130
7.2.1	单元测试考虑	130
7.2.2	单元测试过程	132
7.3	集成测试	134
7.3.1	自顶向下结合	135
7.3.2	自底向上结合	137
7.3.3	不同集成测试策略的比较	137
7.4	验收测试	138
7.4.1	验收测试的范围	138
7.4.2	软件配置复查	139
7.5	设计测试方案	139
7.5.1	逻辑覆盖	139
7.5.2	等价划分	143
7.5.3	边界值分析	146
7.5.4	错误推测	147
7.5.5	实用测试策略	148
7.6	调试	151
7.6.1	调试技术	151
7.6.2	调试策略	152
7.7	软件可靠性	154
7.7.1	基本概念	154
7.7.2	估算平均无故障时间的方法	155
7.7.3	程序正确性证明	157
7.8	日立预测法	158
7.8.1	测试完成率模型	158
7.8.2	错误发现率模型	159
7.8.3	使用日立预测法的步骤	159
7.9	自动测试工具	159
7.9.1	测试数据生成程序	160
7.9.2	动态分析程序	160
7.9.3	静态分析程序	160
7.9.4	文件比较程序	161
7.10	小结	161
	习题七	162
	第八章 维护	166

8.1	软件维护的定义	166
8.2	维护的特点	167
8.2.1	结构化维护与非结构化维护的对比	167
8.2.2	维护的代价	168
8.2.3	维护的问题	168
8.3	维护过程	169
8.3.1	维护组织	169
8.3.2	维护报告	169
8.3.3	维护的事件流	170
8.3.4	保存维护记录	171
8.3.5	评价维护活动	171
8.4	可维护性	172
8.4.1	决定软件可维护性的因素	172
8.4.2	文档	173
8.4.3	可维护性复审	174
8.5	小结	174
	习题八	175
第九章	面向对象方法学引论	176
9.1	传统方法学的缺点	176
9.1.1	存在的问题	176
9.1.2	出现问题的原因	178
9.2	软件工程的新途径	180
9.2.1	快速原型法	180
9.2.2	面向对象方法学	182
9.2.3	两种新途径的结合	187
9.3	基本概念	188
9.3.1	对象	188
9.3.2	其他概念	191
9.4	面向对象建模	194
9.5	对象模型	196
9.5.1	表示类—&—对象的图形符号	196
9.5.2	表示结构的图形符号	197
9.5.3	例子	200
9.6	动态模型	201
9.6.1	术语	201
9.6.2	表示方法	202
9.6.3	例子	202
9.7	功能模型	203

9.7.1	表示方法	203
9.7.2	与其他两种模型的关系	203
9.8	小结	204
习题九		205
第十章	面向对象分析	206
10.1	面向对象分析的基本过程	206
10.1.1	概述	206
10.1.2	三个子模型与五个层次	207
10.2	需求陈述	208
10.2.1	书写要点	208
10.2.2	例子	209
10.3	建立对象模型	210
10.3.1	确定类—&—对象	210
10.3.2	确定关联	212
10.3.3	划分主题	216
10.3.4	确定属性	216
10.3.5	识别继承关系	219
10.3.6	反复修改	220
10.4	建立动态模型	221
10.4.1	编写脚本	222
10.4.2	设想用户界面	223
10.4.3	画事件跟踪图	224
10.4.4	画状态图	224
10.4.5	审查动态模型	227
10.5	建立功能模型	227
10.5.1	画出基本系统模型图	227
10.5.2	画出功能级数据流图	228
10.5.3	描述处理框功能	228
10.6	定义服务	229
10.6.1	常规行为	229
10.6.2	从事件导出的操作	229
10.6.3	与数据流图中处理框对应的操作	230
10.6.4	利用继承减少冗余操作	230
10.7	小结	230
习题十		230
第十一章	面向对象设计	232
11.1	面向对象设计的准则	232
11.1.1	模块化	233

11.1.2	抽象	233
11.1.3	信息隐藏	233
11.1.4	弱耦合	233
11.1.5	强内聚	234
11.1.6	可重用	234
11.2	启发规则	235
11.2.1	设计结果应该清晰易懂	235
11.2.2	一般—特殊结构的深度应适当	235
11.2.3	设计简单的类	235
11.2.4	使用简单的协议	236
11.2.5	使用简单的服务	236
11.2.6	把设计变动减至最小	236
11.3	软件重用	237
11.3.1	概念	237
11.3.2	软件重用的效果	237
11.3.3	软件重用技术	239
11.3.4	类构件	239
11.4	系统分解	241
11.4.1	子系统之间的两种交互方式	242
11.4.2	组织系统的两种方案	242
11.4.3	设计系统的拓扑结构	243
11.5	设计问题域子系统	243
11.5.1	调整需求	244
11.5.2	重用已有的类	244
11.5.3	把问题域类组合在一起	244
11.5.4	增添一般化类以建立协议	244
11.5.5	调整继承层次	245
11.5.6	ATM 系统之例	247
11.6	设计人-机交互子系统	247
11.6.1	设计人-机交互界面的准则	247
11.6.2	设计人-机交互子系统的策略	248
11.7	设计任务管理子系统	250
11.7.1	分析并发性	250
11.7.2	设计任务管理子系统	250
11.8	设计数据管理子系统	252
11.8.1	选择数据存储管理模式	252
11.8.2	设计数据管理子系统	253
11.8.3	例子	255

11.9	设计类中的服务	255
11.9.1	确定类中应有的服务	255
11.9.2	设计实现服务的方法	256
11.10	设计关联	257
11.10.1	关联的遍历	257
11.10.2	实现单向关联	257
11.10.3	实现双向关联	257
11.10.4	链属性的实现	258
11.11	设计优化	258
11.11.1	确定优先级	258
11.11.2	提高效率的几项技术	259
11.11.3	调整继承关系	260
11.12	小结	262
	习题十一	263
第十二章	面向对象实现	264
12.1	程序设计语言	264
12.1.1	面向对象的语言与非面向对象的语言	264
12.1.2	面向对象语言的技术特点	265
12.1.3	选择面向对象语言	268
12.2	程序设计风格	269
12.2.1	提高可重用性	269
12.2.2	提高可扩充性	272
12.2.3	提高健壮性	272
12.3	面向对象测试	273
12.4	小结	274
	习题十二	275
第十三章	管理技术	276
13.1	成本估计	276
13.1.1	参数方程	276
13.1.2	标准值法	277
13.1.3	COCOMO 模型	279
13.2	进度计划	282
13.2.1	Gantt 图(横道图)	282
13.2.2	工程网络	283
13.2.3	估算进度	285
13.2.4	关键路径	286
13.2.5	机动时间	286
13.3	人员组织	288

13.3.1	程序设计小组的组织	289
13.3.2	主程序员组	289
13.4	质量保证	290
13.4.1	软件质量	290
13.4.2	质量保证	292
13.5	项目计划	293
13.5.1	项目计划的内容	293
13.5.2	项目报告	293
13.5.3	变动控制	294
13.6	软件管理工具	295
13.7	小结	295
	习题十三	295
附录 A C++类库管理系统的分析与设计		297
A.1	面向对象分析	297
A.1.1	需求	297
A.1.2	建立对象模型	298
A.2	面向对象设计	299
A.2.1	设计类库结构	299
A.2.2	设计问题域子系统	300
A.2.3	设计人-机交互子系统	301
A.2.4	设计其他类	303
附录 B 一个汉字行编辑程序的设计		305
B.1	设计规格说明	305
B.1.1	外部编辑命令	305
B.1.2	编辑命令	306
B.1.3	输出信息	306
B.2	概要设计	308
B.2.1	正文文件	308
B.2.2	两个工作模式	309
B.2.3	数据元素	310
B.2.4	过程	310
B.3	概要设计结果	311
B.4	详细设计	314
B.4.1	数据元素	314
B.4.2	控制数据元素	316
B.4.3	编辑过程	316
B.4.4	输入模式的过程	317
B.4.5	编辑模式的过程	318