



21世纪复旦大学研究生教学用书

软件工程：方法与实践

赵文耘 彭 鑫 张 刚 沈立炜 著

复旦大学出版社



21世纪复旦大学研究生教学用书

软件工程：方法与实践

赵文耘 彭 鑫 张 刚 沈立炜 著

复旦大学出版社

图书在版编目(CIP)数据

软件工程:方法与实践/赵文耘等著. —上海:复旦大学出版社,2014.12
21世纪复旦大学研究生教学用书
ISBN 978-7-309-11010-4

I. 软… II. 赵… III. 软件工程-高等学校-教学参考资料 IV. TP311.5

中国版本图书馆CIP数据核字(2014)第230231号

软件工程:方法与实践

赵文耘 彭鑫 张刚 沈立炜 著
责任编辑/梁玲

复旦大学出版社有限公司出版发行
上海市国权路579号 邮编:200433
网址:fupnet@fudanpress.com <http://www.fudanpress.com>
门市零售:86-21-65642857 团体订购:86-21-65118853
外埠邮购:86-21-65109143
常熟市华顺印刷有限公司

开本 787×1092 1/16 印张 24.25 字数 561 千
2014年12月第1版第1次印刷

ISBN 978-7-309-11010-4/T·525

定价:59.00元

如有印装质量问题,请向复旦大学出版社有限公司发行部调换。
版权所有 侵权必究

作者简介

赵文耘,复旦大学软件学院教授、博士生导师。1984年毕业于复旦大学计算机科学系,1989年获得硕士学位。从1984年开始在复旦大学任教至今,曾任复旦大学档案馆副馆长、复旦大学校园网管理服务中心主任,现任复旦大学软件工程实验室主任、软件工程一级学科学术带头人、软件工程博士后流动站站长。学术兼职包括中国计算机学会软件工程专业委员会副主任、中国软件行业协会软件工程分会副理事长。研究方向为软件工程、软件开发工具及其环境、企业应用集成(EAI)。自1989年到2000年一直参加国家科技攻关项目“大型软件开发环境青鸟系统”的研制和开发,为复旦大学所承担的各子专题的主要开发人员或子专题负责人,该系统获1998年国家科技进步二等奖、1996年电子工业部科技进步特等奖和1996年三部委联合颁发的荣誉证书。2001年至今参加的863项目“基于Internet以构件库为核心的软件开发平台”和“上海构件库及其应用研究”分别获得2006年国家科技进步二等奖和2005年上海市科技进步二等奖。主持上海市精品课程“软件工程”,获得“复旦大学教学名师”称号,负责的“软件工程课程教学与实践”获2009年上海市教学成果奖二等奖。

彭鑫,博士,复旦大学软件学院副教授、博士生导师。现为中国计算机学会高级会员、软件工程专业委员会委员、上海市计算机学会青年工作委员会委员。2001年毕业于复旦大学计算机科学与工程系,获学士学位。2006年复旦大学计算机软件与理论专业研究生毕业,获博士学位。2010年晋升为副教授,2013年入选复旦大学“卓学计划”。主要研究方向包括软件维护与演化、软件产品线、自适应软件、移动计算与云计算等。至今已在国际会议及国内外期刊上发表论文40余篇,并担任了ICSR, SEKE, ICSM等国际会议的程序委员会委员。研究工作获得第27届软件维护国际会议最佳论文奖。

张刚,博士、高级工程师,上海理工大学光电信息与计算机工程学院教师、硕士生导师。研究方向为软件工程,包括软件开发方法、软件体系结构、软件维护等,在国际会议上发表多篇论文。2013年毕业于复旦大学计算机科学技术学院,获博士学位。此前在上海贝尔任职12年,历任软件工程师、高级工程师、软件开发经理、主任工程师和顾问工程师等职,曾长期担任全球第一款全IP千兆接入产品ISAM7302的总体软件架构师。自2009年起担任公司战略与转型部门技术教练,成功引领了多个产品和团队的敏捷转型。2013年被贝尔实验室授予“杰出工程师”称号。

沈立炜,博士,复旦大学软件学院讲师。2009年毕业于复旦大学计算机科学技术学院,获博士学位,博士期间曾前往美国北卡罗莱纳大学夏洛特分校进行学术访问。目前主要从事软件工程方面的教学与科研工作,研究方向包括软件体系结构、软件产品线及自适应软件等。参加多项国家863项目与自然科学基金项目的研究,在国内外相关领域学术会议和期刊上发表论文10余篇。

推动科技出版事业
提高学术水平

为「上海科技专著出版资金」题

徐匡迪

二〇〇〇年十月十日

科学技术是第一生产力。21 世纪,科学技术和生产力必将发生新的革命性突破。

为贯彻落实“科教兴国”和“科教兴市”战略,上海市科学技术委员会和上海市新闻出版局于 2000 年设立“上海科技专著出版资金”,资助优秀科技著作在上海出版。

本书出版受“上海科技专著出版资金”资助。

上海科技专著出版资金管理委员会

编辑出版说明



21 世纪,随着科学技术的突飞猛进和知识经济的迅速发展,世界将发生深刻变化,国际间的竞争日趋激烈,高层次人才的教育正面临空前的发展机遇与巨大挑战。

研究生教育是教育结构中高层次的教育,肩负着为国家现代化建设培养高素质、高层次创造性人才的重任,是我国增强综合国力、增强国际竞争力的重要支撑。为了提高研究生的培养质量和研究生教学的整体水平,必须加强研究生的教材建设,更新教学内容,把创新能力和创新精神的培养放到突出位置上,必须建立适应新的教学和科研要求的有复旦特色的研究生教学用书。

“21 世纪复旦大学研究生教学用书”系列教材正是为适应这一新形势而编辑出版的。该系列分文科、理科和医科三大类,主要出版硕士研究生学位基础课和学位专业课的教材,同时酌情出版一些使用面广、质量较高的选修课及博士研究生学位基础课教材。这些教材除可作为相关学科的研究生教学用书外,还可以供有关学者和人员参考。

收入“21 世纪复旦大学研究生教学用书”的教材,大都是作者在编写成讲义后,经过多年教学实践、反复修改后才定稿的。这些作者大都治学严谨,教学实践经验丰富,教学效果也比较显著。由于我们对编辑工作尚缺乏经验,不足之处,敬请读者指正,以便我们在将来再版时加以更正和提高。

复旦大学研究生院

前言

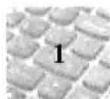


随着计算机和软件技术的发展,计算机软件正越来越深入地影响并改变着人们工作和生活的方方面面。与此同时,计算机软件的开发、维护和运行也面临着质量不可靠、开发效率低、维护困难等诸多问题和挑战。软件工程是一门研究用工程化方法构建和维护有效、实用和高质量软件的学科,涉及软件需求、软件设计、程序设计、软件测试、软件维护等多个方面,是实现工程化的软件开发、维护和运行的重要基础。自从20世纪60年代“软件工程”的概念被提出之后,软件工程相关的方法、技术和工具等都有了长足的进步。与此同时,工业界的软件工程实践也有了很大的发展,积累了很多重要的工程经验和最佳实践。

在总结、凝练软件工程研究和实践的基础上,许多国内外学者和出版机构都出版了介绍软件工程方法、技术和实践的著作。这些著作以软件工程的经典开发方法,如瀑布模型、结构化开发方法、面向对象开发方法等为基础,以普及和介绍软件工程的基本概念和思想为主要目的。

近年来,随着互联网的迅猛发展以及应用形态和软件技术的不断创新,围绕着基于互联网的软件开发、维护和运行,软件工程领域发生了一系列重要的突破性进展,也产生了许多新的软件工程思想、方法和技术。例如,敏捷开发、基于构件的软件开发方法、软件产品线开发方法等已经得到了广泛应用;一些有效的软件开发技术和实践,如契约式设计、重构、模型驱动的体系结构等也逐渐普及;同时,围绕大规模、复杂软件系统开发及维护的支持工具和技术,如版本管理、自动化测试、程序理解、逆向工程等,也受到了越来越广泛的关注。目前还缺少一本全面介绍这些新的思想、方法和技术的软件工程著作。

本书的主要目的是为软件工程领域的研究和实践者提供一本反映近年来软件开发方法、技术、工具和实践方面最新进展以及未来发展方向和趋势的专业著作。它需要既能保证内容的全面性、覆盖软件工程领域的基本概念和思想,又能突出软件工程研究和实践的最新进展,这是一项富有挑战性的工作。本书主要按照软件开发生存周期的过程和阶段进行组织,然后对一些软件开发的新方法和新技术进行介绍。第1章对软件工程的基本概念和根本性困难进行了介绍和分析,第2至第7章分别介绍软件过程、需求工程、软件设计、软件构造、软件测试和软件维护。这部分内容突出了方法的系统性及实践背后的基本原理,同时对



于相关的软件工具和实践指南进行了扼要介绍。在此基础上,我们还介绍了一些新的软件开发方法。其中,第8章介绍了软件复用与构件技术,第9章介绍了软件产品线,第10章介绍了面向方面的软件开发等软件开发新技术。

赵文耘编写了本书的第1章和第8章,并负责全书的统稿;彭鑫编写了本书的第3章、第7章和第9章;张刚编写了本书的第2章、第4章和第5章;沈立炜编写了本书的第6章和第10章。此外,陈碧欢、董瑞志、林云、钱文亿等研究生也参加了相关资料的收集和整理工作。

本书的出版受到了复旦大学研究生院资助项目以及上海科技专著出版资金的支持,同时也得到了复旦大学软件学院的大力支持,在此一并表示感谢。此外,我们还要感谢复旦大学出版社的黄乐、梁玲等编辑的信任和支持。

我们希望这本书能够为软件工程领域的研究者了解相关方面的研究状况并开展相关研究工作打下基础,同时为项目经理、架构师等高级工程技术人员掌握软件开发过程管理、软件设计和构造技术等方面的前沿技术和最佳实践提供帮助。

编者

2014年7月



目

录

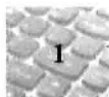


第 1 章

软件工程概述	1
1.1 软件工程基本概念	1
1.2 软件开发的根本性困难	2
1.3 软件工程知识领域	3
1.4 各章内容简介	4
1.4.1 软件过程	4
1.4.2 需求工程	5
1.4.3 软件设计	6
1.4.4 软件构造	6
1.4.5 软件测试	7
1.4.6 软件维护	7
1.4.7 软件复用与构件技术	8
1.4.8 软件产品线	9
1.4.9 软件开发新技术	9
本章参考文献	10

第 2 章

软件过程	11
2.1 概述	11
2.1.1 基本概念	11
2.1.2 ISO/IEC 12207 软件生存周期过程	12
2.1.3 软件过程模型	17
2.2 统一软件开发过程	20
2.2.1 迭代和增量的过程框架	21
2.2.2 用况驱动,以构架为中心	23
2.2.3 核心工作流	24
2.3 敏捷和精益方法	27
2.3.1 概述	27
2.3.2 敏捷方法的价值观和原则	28
2.3.3 精益思想	31
2.3.4 敏捷和精益实践简介	35



2.4 软件能力成熟度和过程改进	42
2.4.1 软件过程改进	42
2.4.2 能力成熟度模型集成	44
本章参考文献	46

第3章

软件需求工程	48
3.1 需求工程概述	48
3.1.1 软件需求	48
3.1.2 现状与挑战	49
3.1.3 需求制品	50
3.1.4 需求工程过程及活动	51
3.2 系统与上下文分析	52
3.2.1 软件密集型系统	52
3.2.2 从系统需求到软件需求	54
3.2.3 系统上下文	55
3.2.4 问题框架方法	56
3.3 需求制品	61
3.3.1 目标	61
3.3.2 场景	67
3.3.3 面向方案的需求	74
3.4 需求工程活动	78
3.4.1 需求获取	78
3.4.2 需求分析	80
3.4.3 文档化	81
3.4.4 需求确认	82
3.5 需求管理	83
3.5.1 需求追踪管理	83
3.5.2 需求优先级管理	85
3.5.3 需求变更管理	86
本章参考文献	87

第4章

软件设计	89
4.1 软件设计概述	89
4.1.1 软件设计的目标	89
4.1.2 软件设计的相关概念	90
4.1.3 软件设计的设计原则	91
4.1.4 软件设计的过程模型	93
4.2 模型和视图	95

4.2.1	模型	95
4.2.2	视图	95
4.2.3	UML 建模语言	96
4.3	质量属性	99
4.3.1	质量属性和设计	99
4.3.2	ISO/IEC 25010	100
4.3.3	质量属性效用树	102
4.4	软件体系结构设计	104
4.4.1	软件体系结构的定义	104
4.4.2	软件体系结构视图	106
4.4.3	风格、模式和框架	111
4.4.4	软件体系结构模式	112
4.5	设计评审	119
4.5.1	设计评审目标	119
4.5.2	设计评审原则	120
4.5.3	体系结构评估方法	121
	本章参考文献	126

第5章

软件构造	128
5.1 软件构造的概述	128
5.2 构造阶段的设计	129
5.2.1 抽象数据类型	130
5.2.2 类和接口	131
5.2.3 依赖	135
5.2.4 契约式设计	138
5.3 编码	141
5.3.1 语言	141
5.3.2 编程规范	144
5.3.3 错误处理	145
5.4 单元测试	147
5.4.1 基本概念	148
5.4.2 自动化单元测试框架	148
5.4.3 依赖和测试替身	150
5.4.4 测试先行	151
5.4.5 测试驱动开发	152
5.5 集成	153
5.5.1 “大爆炸”集成和增量集成	153
5.5.2 持续集成	153

5.6 软件构造工具	156
5.6.1 版本管理系统	156
5.6.2 编译和调试工具	157
5.6.3 集成开发环境	157
本章参考文献	158

第6章

软件测试	159
6.1 软件测试概述	159
6.1.1 软件测试的价值	159
6.1.2 软件测试的概念	160
6.1.3 典型的软件测试级别	161
6.1.4 软件测试的原则	161
6.2 软件测试过程	162
6.2.1 软件测试过程模型	162
6.2.2 软件测试标准	164
6.3 软件测试技术	164
6.3.1 黑盒测试	164
6.3.2 白盒测试	169
6.3.3 组合测试	175
6.3.4 变异测试	176
6.3.5 静态测试	177
6.4 软件测试类型	178
6.4.1 单元测试	178
6.4.2 集成测试	180
6.4.3 系统测试	183
6.4.4 验收测试	187
6.4.5 回归测试	188
6.5 软件测试工具	189
6.5.1 静态代码分析工具	189
6.5.2 系统测试工具	192
6.6 面向对象软件的测试	194
6.6.1 面向对象软件测试的难点	194
6.6.2 面向对象软件的测试方法	195
本章参考文献	198

第7章

软件维护	200
7.1 软件维护概述	200
7.1.1 软件维护类型	200

7.1.2	软件的可维护性	201
7.1.3	软件维护过程	201
7.1.4	软件再工程	202
7.1.5	软件维护技术	204
7.2	软件分析	204
7.2.1	静态分析	205
7.2.2	动态分析	210
7.2.3	开发历史分析	211
7.3	程序理解与变更影响分析	211
7.3.1	特征定位	212
7.3.2	软件制品追踪关系	212
7.3.3	变更影响分析	213
7.4	软件逆向工程	214
7.4.1	软件逆向工程概述	214
7.4.2	程序度量	216
7.4.3	模型逆向恢复	217
7.4.4	软件体系结构逆向恢复	218
7.4.5	软件可视化	223
7.5	软件重构	225
7.5.1	代码的坏味道	225
7.5.2	基本的软件重构类型	228
7.5.3	重构的原则	231
7.6	软件维护工具	232
7.6.1	缺陷跟踪管理工具	232
7.6.2	任务管理工具	233
7.6.3	特征定位工具	234
7.6.4	克隆分析工具	235
7.6.5	逆向分析工具	236
7.6.6	代码分析和度量工具	237
	本章参考文献	239

第8章

软件复用与构件技术	241
8.1 软件复用基础	241
8.1.1 软件复用分类	241
8.1.2 软件复用的发展、现状与挑战	243
8.1.3 软件复用经济学	245
8.2 软件复用过程	247
8.2.1 软件复用的一般过程	248

8.2.2	复用过程标准 IEEE 1517	249
8.2.3	软件复用成熟度模型	250
8.3	软件复用技术	251
8.3.1	生成式复用	251
8.3.2	组装式复用	256
8.3.3	面向对象技术中的软件复用	257
8.4	基于构件的软件开发	260
8.4.1	软件构件的定义	260
8.4.2	软件构件模型	261
8.4.3	构件组装技术	267
8.4.4	商用成品构件	274
8.4.5	构件管理	276
8.4.6	构件化对企业的影响	279
8.5	基于框架与中间件的复用	280
8.5.1	框架技术	280
8.5.2	中间件技术	285
	本章参考文献	288

第9章

软件产品线	291
9.1 软件产品线基本思想和方法	291
9.1.1 从软件复用到软件产品线	291
9.1.2 软件产品线工程	292
9.1.3 软件产品线的特点和优势	293
9.1.4 软件产品线开发过程	294
9.2 软件产品线范围和可变性	296
9.2.1 产品线开发经济效益	296
9.2.2 产品线范围	297
9.2.3 软件产品线可变性	300
9.3 领域工程	305
9.3.1 领域需求工程	306
9.3.2 领域设计	313
9.3.3 领域实现	325
9.4 应用系统工程	330
9.4.1 应用系统需求工程	330
9.4.2 应用系统设计	333
9.4.3 应用系统实现	335
9.5 软件产品线管理	336
9.5.1 软件产品线管理中的特殊问题	336



9.5.2 软件产品线组织管理	337
9.5.3 软件产品线技术管理	338
本章参考文献	340

第 10 章

软件开发新技术	342
10.1 面向方面的编程	342
10.1.1 AOP 概述	342
10.1.2 AOP 与 OOP 的关系	345
10.1.3 AOP 的实现框架与工具	346
10.1.4 AspectJ 简介及其应用实例	347
10.1.5 Spring AOP 简介及其应用实例	349
10.1.6 JBoss AOP 简介及其应用实例	353
10.2 面向特征的编程	354
10.2.1 FOP 概述	354
10.2.2 FOP 语言模型	356
10.2.3 FeatureIDE	359
10.3 模型驱动的体系结构	362
10.3.1 MDA 概述	362
10.3.2 MDA 模型体系	364
10.3.3 基于 MDA 的软件开发过程	365
10.3.4 MDA 应用实例	368
本章参考文献	371





软件工程概述

本章将首先介绍软件工程的基本概念,并对软件开发的根本性困难进行分析,然后介绍软件工程知识领域,最后对后续各章的内容进行简要介绍。

1.1 软件工程基本概念

软件工程(Software Engineering)的概念是在 1968 年北大西洋公约组织(NATO)的一次会议上正式提出的。当时工业界所面临的一个普遍问题是:一方面所开发的软件系统的规模和复杂性越来越高(如 IBM 的 OS/360 系统),另一方面手工作坊式的软件开发方式在开发效率、成本控制、质量保障等方面越来越难以满足需要。许多软件开发项目所遭遇的情形正如 Brooks 在其著作《人月神话》中所形容的^[1]:“……正像一只逃亡的野兽落到泥潭中做垂死的挣扎,越是挣扎,陷得越深,最后无法逃脱灭顶的灾难。……程序设计工作正像这样一个泥潭,……一批批程序员被迫在泥潭中拼命挣扎,……谁也没有料到问题竟会陷入这样的困境……”这一问题被形容为“软件危机”,具体表现包括开发进度难以预测、开发成本难以控制、用户对产品功能难以满意、产品质量难以保证、软件难以维护。

国际标准化组织 ISO/IEC/IEEE 在所发布的系统和软件工程术语标准中将软件工程定义如下^[2]:将科学和技术的知识、方法和经验,系统化地应用于软件的设计、实现、测试和文档化;将系统化的、规范的、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中。

软件工程的目的是在开销合理(时间、成本)的情况下开发高质量的软件产品。其中,软件质量关注于软件产品满足预期需求的程度,而开销合理是指软件开发、运行的整个进度和成本满足用户要求的程度。软件工程的基本思想是将工程化的思想、方法和原则应用到软件的开发、维护和运行过程中。例如,软件工程强调量化管理、规范化的开发过程、详细的开发文档、自动化工具的应用等,希望通过严格的工程化开发实现软件开发时间、成本和质量可预测、可控制以及成功经验可复制。

软件工程的基本内容可以概括为一种层次化结构,如图 1-1 所示^[3]。其中,质量关注点(Quality Focus)是软件工程的根基,是工程化软件开发所追求的目标;过程(Process)是软件工程的基础,为工程化的软件开发指明了路线图;方法(Methods)为软件开发过程中遇到的各种问题



图 1-1 软件工程的几个层次^[3]

