

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
软件工程

软件工程导论

朱少民 编著

清华大学出版社



清华大学出版社

清华大学出版社
北京 100084

软件工程导论

第四版

清华大学出版社

高等学校教材
软件工程

软件工程导论

朱少民 编著

清华大学出版社
北京

内 容 简 介

本书围绕软件工程最基本的核心——“质量和生产率”展开,力求通过一些具体的实例来说明软件工程的思想和方法。本书在回答为什么要研究软件工程问题之后,首先系统地给出软件工程的知识体系框架,帮助读者建立起软件工程的完整且清晰的概念。然后深入软件过程、软件工程目标和要求的讨论,全面阐述了软件工程的思想和方法、技术和工具,结合软件工程环境、软件项目管理等内容,使读者坚实地掌握和运用软件工程的内涵。

本书内容丰富,构思新颖,突破软件工程图书的传统框架,使整个软件工程系更清晰,从思想和管理出发,自然延伸到流程、方法、工具和环境,重新阐释了软件工程系。本书吸收了软件工程的最新进展,适应软件即服务(SaaS)的开发模式,增加了相应内容,符合当今软件工程的实际需求。

本书系统性强、通俗易懂,内容丰富且实用,理论和实践有效地结合,可作为高等学校的软件工程专业、计算机软件专业和相关专业的教材,也适合各类软件工程项目管理人员和技术人员学习。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件工程导论/朱少民编著. —北京:清华大学出版社,2009.1

(高等学校教材·软件工程)

ISBN 978-7-302-18623-6

I. 软… II. 朱… 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(2008)第 147368 号

责任编辑:丁 岭 顾 冰

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京嘉实印刷有限公司

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185×260 印 张:17.75 字 数:442 千字

版 次:2009 年 1 月第 1 版 印 次:2009 年 1 月第 1 次印刷

印 数:1~4000

定 价:29.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系
退换。联系电话:(010)62770177 转 3103 产品编号:027763-01

出版说明

高等学校教材·软件工程

改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制订的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制订精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻

性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

(6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过 20 多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业作出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

前言

高等学校教材·软件工程

40年前,人们就开始讨论“软件工程”这样一个话题,但至今软件工程依然不太成熟,例如今天的软件质量水平依旧不高,软件的开发模式还在探索之中,而这一切主要归结于软件技术的日新月异的变化和软件自身的复杂特性。互联网的普及将软件技术的变化推向新的高潮,人们借助互联网的力量可以随时随地沟通、协作,可以共享知识、技能和经验,甚至可以积聚全世界的力量共同探讨同一个技术主题,所有这些极大地推动软件技术的发展。而在这同时,软件产业也在悄悄地发生着巨大的变化,从传统的软件产品销售模式向软件服务模式转化,软件即服务(software as a service)或按需服务(on-demand service)的趋势越来越明显,其中最具代表性的服务就是 Salesforce。Salesforce 为中小型企业提供各种业务应用的在线服务,每年以 80% 的速度增长,客户满意度高达 97%,从销售团队自动化到合作伙伴关系管理、市场营销和客户服务, Salesforce 重新定义客户关系管理。企业不再需要部署自己的服务器、不需要购买软件等,只要按照自己的实际需求,访问 Salesforce.com 以获得自己所需要的业务处理功能,每个月或每年只要付出很低的服务费。软件开发模式,也自然随着 SaaS 模式诞生而正在发生巨大的变化,有必要在这关键时刻重新审视软件工程的思想、方法和实践,这也是本书写作的主要理由。

软件工程不仅面临着技术突飞猛进的挑战,还要面临需求变化频繁、质量难以控制的巨大挑战。下面两个例子,在某种程度上说明了这种挑战的严峻性。

例一：交通红绿灯的需求变化。

大家都非常熟悉街道上的红绿灯,可以根据不同方向的车流量和人流,进行调节,实施智能控制。但早期的红绿灯,可不是这样的,非常简单,一个方向红的时候则另一个方向绿,每隔 60 秒交替变化,这时用简单的定时模拟电路控制就可以了。

后来,人们觉得这样的设计不够人性化,中途经过十字路口的行人或驾驶员不知道要等多长时间,绿灯才会亮。所以,加上一个数字计时器,显示剩下的等待秒数。这时,模拟电路控制就必须改为数字电路,需要重新设计和实现。

再到后来,人们发现一个十字路口的两个方向,车与人的流量是不一样的,需要调整不同方向的红绿灯切换的间隔时间,例如一个方向是 75 秒而另外一个方向是 25 秒。而且,将来的车流量会发生变化,即红绿灯间隔时间用户可自行设置,这样要求对原来的数字电路或控制程序进行修改,这种改动可能很大,需要修改设计和修改实现。

例二：简单讽刺软件工程的现状。

- (1) 程序员写出自认为没有 Bug(缺陷)的代码。
- (2) 软件测试,发现了 100 个 Bug。
- (3) 程序员修改了 50 个 Bug,并告诉测试组另外 50 个 Bug 不是 Bug。
- (4) 在已修正的 50 个 Bug 中,测试组验证时,发现其中 20 个仍然存在,同时又发现了 30 个新 Bug。
- (5) 不断重复上面的步骤(3)和步骤(4)。
- (6) 鉴于市场方面的压力,为了配合当初制定的过分乐观的发布时间表,产品还是按时上市了。
- (7) 用户发现了不少问题(近百个 Bug),反馈到研发部。
- (8) 已经领了项目奖金的程序员不知跑到哪里去了。
- (9) 新组建的项目组差不多修正了全部 Bug,但测试组又发现了 80 多个 Bug。
- (10) 早先离开的程序员打电话给测试组,将他们挖走。
- (11) 公司的软件发布更快了,因为现在是开发人员自己来测试自己写的程序,发现的缺陷很少。
- (12) 客户发现的问题越来越多,抱怨越来越多。
- (13) 客户开始减少得很快,公司很快倒闭。
- (14) 新的公司被组建,新进来的程序员写出自认为没有 Bug 的代码。

软件需求总是变化的,这种变化来源于客户需求的变化,其中许多变化不是用户驱动的而是由竞争对手驱动的。许多需求的变化导致产品架构变化,原先的设计和实现不能适应这种变化,就必须重新设计和重新实现,这就是重构。软件的迭代开发或重构,正是适应这种特定的需求,并日益受到重视,最终导致软件工程思想和方法的变化。

软件质量的改善,已迫在眉睫,招聘大量的测试人员可以部分地解决问题,但不能彻底地解决问题,因为质量是构建出来的,而不能靠测试测出来。当软件中存在大量的缺陷,虽然经过充分的测试,但软件发布时漏掉的缺陷可能还会不少,而且测试、开发人员的返工引起的代价很大。所以,要真正提高质量,要将需求、设计和编码等各项工作做好,归纳起来,做每项工作的时候,第一次就把它做对,这就是缺陷预防的思想。

本书在交代了软件危机、软件过程内容、软件工程目标和要求等基础之上,强调建立正确的软件工程思想,思想是万物之源,思想会决定流程和方法。另一方面,思想需要借助特定的方法来实现,而方法需要付诸于实践、由实践来检验。这就是本书的基本构思,从思想到方法,从方法到建模,再到软件环境、工具,逐步向前推进,不断揭示软件工程的内涵。通过软件工程的思想、方法、技术和工具的全面介绍,帮助读者了解完整的软件工程系,为将来深入地学习需求工程、软件设计、软件测试等课程打下坚实的基础。通过下表,可以更清楚本书的写作思路,更好地学习本书的内容。

问 题	章节、内容
<ul style="list-style-type: none"> 什么是软件工程 为什么要讨论软件工程 	1.1 软件危机 1.2 软件的问题在哪里 1.3 软件工程的诞生 1.4 软件工程的命题
<ul style="list-style-type: none"> 软件工程包含哪些内容 在软件工程中要学习哪些知识 	1.5 软件工程知识体系
<ul style="list-style-type: none"> 过去几十年,软件工程发生了哪些变化 为什么会发生这些变化 	1.6 现代软件工程
<ul style="list-style-type: none"> 如何理解软件工程 成功的软件工程是怎样的 	2.1 完整的软件生命周期 2.2 需求工程 2.3 设计 2.4 实施 2.5 部署、运行和维护 2.6 软件非工程过程
<ul style="list-style-type: none"> 软件工程带来的益处又是什么 软件工程的基本思想是什么 	3.1 软件工程的基本目标 3.2 软件工程的影响要素 3.3 软件工程的业务需求 3.4 软件工程的质量要求 3.5 软件工程的成本要求 3.6 软件工程的资源限制
<ul style="list-style-type: none"> 软件工程中有哪些要求 软件工程中要考虑的因素有哪些 	4.2 以人为本 4.3 软件开发不是一门艺术 4.4 向传统工业学习 4.5 软件工程的例外 4.6 软件工厂思想 7.1 环境造就软件
<ul style="list-style-type: none"> 如何指导软件工程的开展 软件工程的思想 	第4章 软件工程思想 5.1 软件方法论 5.2 用户需求的获取方法 5.3 软件工程的分析方法 5.4 软件工程的设计方法 5.5 软件测试方法
<ul style="list-style-type: none"> 如何开展软件工程 开展软件工程有哪些方法 	6.2 软件建模 7.2 软件工程组织 7.3 软件工程文化 7.4 软件工程基础设施 8.3 制定计划 8.4 资源管理 8.5 进度和成本管理 8.6 质量管理 8.7 风险管理 8.8 软件配置管理 8.9 项目跟踪和控制
<ul style="list-style-type: none"> 软件工程实施中最关键的是什么 如何保证软件工程实施顺利 	

问 题	章节、内容
• 如何更有效地实施软件工程	6.3 元建模 6.4 建模语言和 UML 9.3 商业工具解决方案 9.4 开源工具集成的解决方案
• 如何改进软件工程 • 如何更好地理解软件过程	6.5.1 瀑布模型的不足 6.5.2 V 模型诠释软件过程 6.5.3 没有统一天下的 RUP 6.5.4 MSF 的过程模型 7.5 过程定义 7.6 过程评估和改进

本书特别重视理论与实践相结合,使读者既能领会软件工程的思想和方法,又能将这些思想、方法等应用到实际工作中去。因此,本书既适合作为计算机软件、软件工程等学科的大学教材,也适合从事软件工程人员阅读,包括软件项目经理、软件开发人员和软件测试人员。

由于作者水平有限,本书难免会存在一些差错,恳请读者及时提出宝贵意见。

作者

2008年6月

目 录

11	11
33	33
36	36
85	85
高等学校教材·软件工程		
38	38
39	39
10	10
11	11
11	11
81	81
引子	1
第 1 章 为什么要研究软件工程	4
1.1 软件危机	5
1.1.1 软件不再只是程序	5
1.1.2 危机四伏	7
1.1.3 软件危机之证	9
1.2 软件的问题在哪里	10
1.2.1 软件特性	10
1.2.2 软件业所面临的问题	12
1.2.3 软件问题的幽默画	13
1.3 软件工程的诞生	14
1.4 软件工程的命题	15
1.5 软件工程知识体系	17
1.5.1 软件工程知识体系的构成	17
1.5.2 软件工程要素	19
1.5.3 SWEBOK	20
1.6 现代软件工程	22
1.6.1 开源软件运动	23
1.6.2 SaaS	24
1.6.3 现代软件工程的基本思想	25
1.7 小结	26
思考题	27
第 2 章 软件过程	28
2.1 完整的软件生命周期	28
2.1.1 软件的工程过程	29
2.1.2 综合的软件过程	32

2.2	需求工程	34
2.2.1	软件需求	35
2.2.2	需求开发	36
2.2.3	需求管理	38
2.3	设计	38
2.3.1	软件设计的目标	38
2.3.2	体系结构设计	39
2.3.3	详细设计	40
2.3.4	部署设计	41
2.3.5	设计评审	41
2.4	实施	42
2.4.1	单元测试	42
2.4.2	集成测试	42
2.4.3	功能测试	43
2.4.4	系统测试	44
2.4.5	验收测试	44
2.5	部署、运行和维护	45
2.5.1	系统部署	46
2.5.2	软件运行和技术支持	46
2.5.3	维护过程	47
2.6	软件非工程过程	47
2.6.1	计划	48
2.6.2	质量保证和管理的过程	50
2.6.3	项目管理过程	51
2.6.4	基础设施和过程改进	53
2.7	小结	54
	思考题	55
第3章	软件工程目标和要求	56
3.1	软件工程的基本目标	56
3.2	软件工程的影响要素	57
3.3	软件工程的业务需求	57
3.4	软件工程的质量要求	58
3.4.1	质量的含义	58
3.4.2	客户是质量的焦点	59
3.4.3	软件质量的特性	60
3.4.4	影响软件产品质量的因素	62
3.4.5	软件过程的质量因素	64
3.4.6	软件质量的指标	65

801	3.5 软件工程的成本要求	68
	3.5.1 软件工程的固有成本	68
901	3.5.2 软件的可变成本	70
1001	3.5.3 软件工程的劣质成本	71
011	3.6 软件工程的资源限制	72
811	3.7 软件产品集成要求	73
111	3.7.1 传统产业的启示	74
211	3.7.2 软件产品工程	74
311	3.8 小结	75
711	思考题	75
811	第4章 软件工程思想	76
051	4.1 永远不可低估思想的作用	76
091	4.2 以人为本	78
131	4.3 软件开发不是一门艺术	79
831	4.4 向传统工业学习	80
731	4.4.1 客户为导向	80
031	4.4.2 需求分析是基础	82
331	4.4.3 过程决定结果	83
631	4.4.4 持续改进过程	84
931	4.4.5 缺陷预防	87
131	4.4.6 围绕项目开展工作	88
231	4.4.7 验证和确认缺一不可	89
331	4.4.8 以架构设计为中心	90
631	4.5 软件工程的例外	90
731	4.5.1 迭代	91
831	4.5.2 敏捷开发思想	93
1381	4.5.3 持续构建和集成	94
1381	4.5.4 永远的 Beta	95
1401	4.5.5 面向对象是一种思想	97
1401	4.5.6 软件工程应归为知识管理	100
1421	4.6 软件工厂思想	102
1441	4.6.1 模块化思想	103
1461	4.6.2 面向构件思想	103
1481	4.6.3 软件复用	105
1481	4.6.4 中间件	105
1481	4.6.5 自动化无处不在	106
1511	4.7 小结	107
1521	思考题	107

第 5 章 软件工程方法	108
5.1 软件方法论	109
5.1.1 系统工程方法论	109
5.1.2 软系统方法论	110
5.1.3 软件工程方法体系	113
5.2 用户需求的获取方法	114
5.3 软件工程的分析方法	115
5.3.1 系统分析方法	115
5.3.2 问题分析方法	117
5.3.3 根本原因分析	118
5.3.4 决策分析	118
5.4 软件工程的设计方法	120
5.4.1 原型设计方法	120
5.4.2 结构化方法	121
5.4.3 面向对象方法	123
5.4.4 面向构件设计方法	127
5.4.5 面向服务方法	129
5.4.6 可视化方法	132
5.5 软件测试方法	133
5.5.1 白盒和黑盒的测试方法	133
5.5.2 静态和动态的测试方法	134
5.5.3 ALAC 测试和随机测试	135
5.5.4 自动化测试方法	135
5.6 小结	136
思考题	137
第 6 章 软件建模	138
6.1 一个简单建模的例子	138
6.2 软件建模	140
6.2.1 为什么要建模	140
6.2.2 软件建模概述	142
6.2.3 如何建模	144
6.3 元建模	145
6.4 建模语言和 UML	147
6.4.1 建模语言的起源	147
6.4.2 实体关系建模	149
6.4.3 UML	151
6.4.4 UML 视图	152

6.4.5	建模机制	155
6.4.6	系统建模语言 SysML	157
6.4.7	虚拟现实建模语言	159
6.5	软件过程模型	162
6.5.1	瀑布模型的不足	162
6.5.2	V 模型诠释软件过程	163
6.5.3	没有统一天下的 RUP	164
6.5.4	MSF 的过程模型	167
6.6	小结	169
	思考题	169
第 7 章	软件工程环境	170
7.1	环境造就软件	170
7.2	软件工程组织	172
7.2.1	软件组织环境	172
7.2.2	软件组织的作用	173
7.2.3	PSP 改进个人能力	175
7.2.4	TSP 提高团队水平	177
7.2.5	完整的组织框架	179
7.3	软件工程文化	180
7.3.1	过程文化	180
7.3.2	质量文化	181
7.4	软件工程基础设施	182
7.5	过程定义	183
7.5.1	软件过程定义的基础	183
7.5.2	剪裁标准软件过程的指南和准则	184
7.6	过程评估和改进	185
7.6.1	软件过程评估的目标和期望	186
7.6.2	软件过程评估的内容和范围	187
7.6.3	软件过程评估的方式和类型	188
7.6.4	软件过程评估的方法	190
7.6.5	过程改进的 IDEAL 模型	191
7.6.6	过程改进的 Raytheon 方法	193
7.6.7	过程改进实施的原则	194
7.6.8	过程改进实施	194
7.7	小结	195
	思考题	196

第 8 章 软件项目管理	197
8.1 软件项目管理概述	197
8.1.1 软件项目管理的 3P	197
8.1.2 软件项目管理的实质	198
8.1.3 软件项目管理的目标和范围	199
8.2 软件项目的分类	200
8.3 制定计划	201
8.3.1 软件规模度量	202
8.3.2 软件开发的估算模型	203
8.3.3 项目工作量估算	204
8.3.4 日程和人力资源安排	207
8.3.5 项目成本估算	209
8.4 资源管理	210
8.5 进度和成本管理	211
8.5.1 里程碑控制和关键路径	212
8.5.2 项目成本控制	213
8.5.3 项目进度的特性及外在关系	214
8.6 质量管理	215
8.6.1 软件质量计划	215
8.6.2 软件评审	216
8.6.3 缺陷预防	218
8.6.4 质量度量	219
8.7 风险管理	221
8.7.1 风险识别	221
8.7.2 风险分析和评估	222
8.8 软件配置管理	223
8.8.1 配置管理的基本概念	223
8.8.2 版本控制	225
8.8.3 变更控制	226
8.9 项目跟踪和控制	227
8.10 小结	228
思考题	229
第 9 章 软件工具	230
9.1 ICASE	230
9.2 IDE	231
9.3 商业工具解决方案	233
9.3.1 典型的完整解决方案	233

9.3.2	商业 IDE	236
9.3.3	需求分析和管理工作	239
9.3.4	建模和设计工具	240
9.3.5	测试工具	241
9.3.6	配置管理工具	242
9.4	开源工具集成的解决方案	243
9.4.1	源代码许可协议	243
9.4.2	开源的 Web 开发框架	245
9.4.3	开源的 IDE	248
9.4.4	PHP 及其他语言的 IDE	249
9.4.5	开源测试工具	250
9.4.6	覆盖软件过程的其他开源工具	253
9.5	小结	255
	思考题	255
附录 A	软件工程术语中英文对照	256
附录 B	项目视图和范围的文档模板	262
参考文献	265