



普通高等教育“十一五”国家级规划教材
软件工程专业核心课程系列教材

丛书主编 孙家广 刘强

软件工程过程

林广艳 姚淑珍 等 编著

- ◆ 教育部高等学校软件工程专业教学指导分委员会推荐教材
- ◆ 根据教育部“软件工程课程体系研究”项目成果《中国软件工程学科教程》及专业规范组织编写
- ◆ 与最新ACM和IEEE CCSE同步
- ◆ 汇集示范性软件工程专业教学成果

清华大学出版社





普通高等教育“十一五”国家级规划教材
软件工程专业核心课程系列教材

丛书主编：孙家广 刘强

软件工程过程

林广艳 姚淑珍 等 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍了软件生存周期过程的基本概念和软件开发过程中包含的关键活动,从历史的角度介绍了软件过程模型的演进历程。通过对已成功地应用于软件企业的 Infosys 模型和协同过程模型的介绍,具体说明了软件生存周期过程中涉及的众多过程、活动与任务间的内在联系以及实施中需要注意的事项。介绍了建立、管理及在优化软件过程中需要考虑的基本问题。

本书是作者多年从事软件开发与管理过程中积累的经验和软件工程教学实践的总结,书中内容力求将较为抽象的软件过程理论与实际应用现状结合,理论联系实际。

本书可作为高等院校软件工程专业及计算机类相关专业的本科和研究生教材,也可供从事软件开发与管理的专业人员参考。

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

软件工程过程/林广艳,姚淑珍等编著. —北京:清华大学出版社,2009.11

(软件工程专业核心课程系列教材)

ISBN 978-7-302-20412-1

I. 软… II. 林… 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 104133 号

责任编辑:丁 岭 李 晔

责任校对:李建庄

责任印制:李红英

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 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×230 印 张:11.25 字 数:247 千字

版 次:2009 年 11 月第 1 版 印 次:2009 年 11 月第 1 次印刷

印 数:1~3000

定 价:20.00 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调换。联系电话:(010)62770177 转 3103 产品编号:023886-01

系列教材编委会

特邀顾问	张尧学					
主 编	孙家广	刘 强				
副 主 编	卢先和					
编 委	陈 钟	梅 宏	张 慧	侯一斌	傅育熙	
	蒋建伟	姚淑珍	刘 超	骆 斌	李宣东	
	臧斌宇	赵一鸣	何钦铭	陈 越	卢 苇	
	李 彤	左保和	刘 琴	曾 明	丁 岭	
策划编辑	丁 岭					

出版说明

近年来,特别是示范性软件学院成立以来,我国软件工程教育事业得到了很大的发展,学科建设逐步走向成熟,初步实现了与国际学科发展的接轨,软件工程教育实现了历史性的跨越。当前,我们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。信息社会不断增长的软件人才需求同教育供给(特别是优质专业教育)不足的矛盾,已成为软件工程教育发展面临的一个重要问题,解决问题的关键之一是需要不断的教学探索之中摸索出一个具有中国特色的软件工程教学思路和方法,形成一个正确的学科体系,在科学的课程体系保证下,使教学质量得到不断提高,使软件工程专业教育得到健康稳步发展。

教育部为此启动了“软件工程课程体系研究”项目(教育部高教司[2003]116号文件),目的是确立软件工程课程体系的框架,确定课程体系的基础及核心内容,并分析预测未来软件工程专业教育的发展趋势,实施精品教材建设。

在孙家广、杨芙清两位院士的积极组织和倡导下,组建了项目课题组。该课题组由两位院士作为主任,集合了清华大学、北京大学、上海交通大学、北京航空航天大学、复旦大学、华中科技大学等高校的一线软件工程教育专家作为成员,并特别聘请了一批软件工程的专家学者作为特邀专家,深入分析 ACM 和 IEEE 发布的《Computing Curriculum-Software Engineering》,研究探讨我国软件工程学科的教育思想、课程体系以及教学计划等,并结合国内部分著名高校软件学院的教学成果,形成了《中国软件工程学科教程 CCSE2004》。此研究成果已成为国内一些大学软件学院制订课程体系的参考依据,并受到普遍关注。

为了进一步推进软件工程专业精品教材的建设,我们在已有研究的基础上,选择了一系列教学基础好、教学内容新、改革成效大、实践素材强的教材组成《软件工程专业核心课程系列教材》,覆盖软件工程的核心知识单元,从而对提高我国软件工程专业的教育水平提供一定的帮助。

软件工程是一门迅速发展的新兴学科,强调理论知识与现实的工程实践相结合。因此,本系列教材在已有软件工程教育知识体系研究的基础上,结合我国软件工程学科的专业发展和现代软件工程人才的培养要求,进一步完善软件工程教育知识体系和核心课程规划,并形成以下特色教材。

(1) 根据 IEEE 最新发布的软件工程知识体系,覆盖软件工程的核心知识单元,整个教

材系列的知识点衔接好。

(2) 全面阐述软件工程理论和实践领域的最新发展,涉及从初始的需求导出到设计和开发以及软件项目管理等软件开发生命周期的各个重要方面,从“可实践”软件工程的角度介绍软件工程的理论、方法与实践。

(3) 从大型复杂软件系统开发的根本问题出发,探讨软件工程的基本原理和方法,论述相关的软件工程实用技术,建立软件开发的系统化和工程化观念。

(4) 以软件工程的最佳实践为基础,分析和比较不同的软件工程实例,避免单纯枯燥的理论讲解,培养学生正确选择和灵活运用这些方法的能力。

(5) 与本教材配套,提供丰富的软件工程资源、案例模型和程序示例,建立软件工程专业核心课程教学网站。

特别感谢教育部软件工程分指导委员会领导和全体委员对本系列教材建设的全力支持与帮助,特别感谢课题组的所有专家,在审稿过程中专家提出了许多中肯的意见,帮助我们严把质量内容,保证丛书的整体质量。

希望在软件工程教学一线的专家同仁根据自身的教学特点,提出宝贵意见(dingl@tup.tsinghua.edu.cn),出版社将大力支持专业领域的教材建设。

对于本系列教材的不足之处,欢迎不吝赐教。

系列教材编委会

2009年10月

总序

“十一五”期间我国的软件工程教育获得了长足发展,专业建设、教学科研水平实现了质的飞跃,在国家政策扶持与众多院校教师、科研人员的共同努力下,逐渐探索出了具有中国特色的软件工程创新人才培养模式与切实可行的创新人才培养方案,核心专业课程建设取得了一定成果。当今数字化、信息化已经成为社会发展的大趋势,高水平的软件工程技术与管理人才队伍在国民经济发展建设特别是信息社会建设中起着重要支撑作用,软件工程专业教育随着技术进步面临着机遇与挑战,培养出色的创新软件科学技术人才成为我们的首要任务。

软件工程精品专业课程与专业教材建设是专业领域建设长抓不懈的重要工作,要通过理论研究、理论创新、教学实践、教材编写、成果推广推动专业课程和教学内容改革,改进教学方法和手段,要通过专业建设建立健全科学的培养体系与教学团队。

为了进一步推进软件工程专业精品教材的建设,物化教学改革实践成果,我们以教育部“软件工程课程体系”项目的研究成果《中国软件工程学科教程》为核心框架,在已有系统地理论研究的基础上,选择了一系列教学基础扎实、教学改革成效显著、实践素材强的教材,不仅能够覆盖软件工程专业知识的核心知识单元,而且能够具有广泛的示范作用,以系列教材的形式推出,完成普通高校“十一五”国家级规划教材建设的光荣任务。

希望这套教材对国内大学的相关学科专业教学具有一定的参考价值和积极的推动作用。希望清华大学出版社能继续大力支持并参与到学院的教学研究、教材建设工作,支持学科的创新与发展。对提高我国软件工程专业的教育水平做出一定的贡献。

软件工程专业、学科是正在迅速发展的综合交叉性学科,相关的新概念、新技术、新方法不断涌现。随着专业、学科的日益成熟与不断创新,教学内容与培养模式的改进与突破迫在眉睫,因此研究和探索专业技术理论与实践发展是十分重要的长期任务,需要我们大家加强交流、共同努力,携手营造良好的教学研究氛围,为推进我国信息科学技术领域教学质量的提高和信息科学技术学科的大发展而不懈努力。

中国工程院院士 孙家广

2009年10月

前言

做任何事情都需要过程,软件开发也不例外。尽管人们不否认过程对软件产品质量的作用,但常常有这样的现象:在进度压力下,首当其冲被简化的工作还是过程。因为规范化的过程不但使过程要素的执行更加严谨,导致短期的活动实施时间拉长,同时需要人力等资源的投入,也无形中增加了软件开发的近期成本。若选择的过程不合适,还会对开发起到阻碍作用。另一方面,任何两个软件产品的开发过程都很难完全相同。而软件过程是对过去经验的总结,因此僵化地照搬任何一个软件过程的结果都可能事与愿违,这也是为什么大家对软件过程敬而远之的原因之一。软件过程的应用与所开发的产品的关键度、参与的人员状况、技术成熟度等因素相关,还与企业的文化相关。过程应用的关键是如何充分体现“平衡”的理念。在一个具体过程中,一个过程或活动是否应该有?若有,应执行到什么样一个“度”?其中的“拿捏”讲究的就是平衡。平衡做得好,则可为软件过程中的各要素创造和谐环境,使项目或产品开发按预期进行;平衡做得不好,则过程各要素间不但相互制约导致预定义过程被束之高阁,其后果是项目或产品的目标也很难达到。软件过程应用中的这些现象,也为弱化软件过程的人们提供了很好的理由。如今交付环境日趋复杂,软件企业已经清醒地认识到:制造软件产品的过程和软件产品本身一样重要。因为高质量的软件产品的背后一定有一个合理的过程来保证。从长远角度看,软件产品的高产出和低成本的背后一定有持续的过程改进做后盾。但如何理解软件工程过程?软件工程过程到底包括哪些内容?如何组织?如何正确地认识软件过程?为什么过程重要却未能得到足够的重视?针对目前软件过程应用中的这些问题或困惑,以及有关软件过程概念的不规范,需要把软件过程的相关内容说清楚,让学生明白软件过程中各要素间的内在联系以及软件过程与企业文化的关系。

为此,作者依据 SWBOK 给出的知识点及多年的软件开发与软件工程教学实践,力求为学生提供一本既满足软件工程本科生教学要求,又能结合实际案例对软件工程过程进行比较全面、清晰地论述的教材。

本书的出版得到了很多人的帮助。感谢麦中凡教授对本书框架提出的非常中肯的建议

并审阅了全书内容。感谢齐治昌教授和杨文龙教授的大力推荐。感谢谭火彬、王华峰、林士媛等为本书编辑、出版所作出的贡献。

因工作做得不够细致,本书中未能明确标记文献的引用。又因时间仓促及水平有限,书中难免有不当之处,敬请读者批评指正。

作 者

2009 年 10 月

目 录

第 1 章 绪论 /1

1.1 软件制造是个复杂的过程	1
1.2 软件产品与软件过程	2
1.3 为什么要研究软件工程过程	4
1.4 软件生存周期过程标准	6
1.5 本书结构	11

第 2 章 软件开发的主要活动 /12

2.1 需求分析与管理	12
2.1.1 需求分析与规范	13
2.1.2 需求变更管理	13
2.1.3 需求跟踪管理	14
2.2 设计	14
2.3 编码	15
2.4 软件测试	15
2.5 运行与维护	16
2.6 软件项目管理	16
2.6.1 项目管理活动	17
2.6.2 软件开发计划	18
2.6.3 风险管理	19
2.7 软件配置管理	20
2.7.1 配置项和基线	20
2.7.2 配置库	22
2.7.3 配置管理流程	23
2.7.4 配置项标识	23

2.7.5	版本控制	25
2.7.6	配置控制	25
2.7.7	状态簿记	26
2.7.8	配置审计	26
2.8	验证与确认	27
2.8.1	V&V 的目标	28
2.8.2	需求分析	28
2.8.3	设计	29
2.8.4	编码	29
2.8.5	测试	30
2.8.6	系统安装试运行	30
2.9	软件质量保证	30
2.9.1	SQA 的作用	30
2.9.2	SQA 启动程序	35
2.9.3	SQA 计划	37
2.9.4	SQA 需要考虑的问题	38
2.10	软件文档管理	39
2.11	本章小结	40

第 3 章 软件生存周期模型 /41

3.1	编码修正模型	42
3.2	瀑布模型	42
3.2.1	瀑布模型的优缺点	43
3.2.2	V 模型	44
3.3	增量模型	45
3.4	演化模型	47
3.5	螺旋模型	48
3.6	原型构造在生存周期模型中的应用	50
3.7	生存周期模型中并发的作用	51
3.8	商业组件和复用的作用	51
3.9	统一软件过程模型(RUP)	52
3.9.1	RUP 过程框架	53
3.9.2	RUP 核心元素	55
3.9.3	RUP 的计划	56
3.9.4	质量内嵌于过程中	56

3.9.5 主要困难与基础保障	57
3.10 MSF 过程模型	57
3.11 本章小结	59

第 4 章 瀑布模型应用实例 /60

4.1 过程实例活动	60
4.1.1 Infosys 过程模型概述	60
4.1.2 需求规范	62
4.1.3 高层设计	66
4.1.4 详细设计	68
4.1.5 构建(编码)与单元测试	69
4.1.6 集成测试计划与实施	70
4.1.7 系统测试计划与实施	72
4.1.8 验收测试与安装	73
4.1.9 维护支持阶段	74
4.2 文档编制	75
4.3 WAR 系统开发过程实施案例	75
4.3.1 WAR 系统需求概述	76
4.3.2 高层设计阶段	85
4.3.3 详细设计阶段	87
4.3.4 构建与单元测试阶段	88
4.3.5 系统测试	89
4.3.6 验收和安装	94
4.4 本章小结	95

第 5 章 协同过程模型 /96

5.1 模型概述	96
5.2 实际应用案例需求	97
5.3 初始阶段	98
5.3.1 基本活动	98
5.3.2 实施考虑	100
5.4 细化阶段	109
5.4.1 基本活动	109
5.4.2 实施考虑	113
5.5 构造阶段	125

5.5.1	主要活动	126
5.5.2	实施考虑	128
5.6	移交阶段	135
5.6.1	基本活动	136
5.6.2	实施考虑	137
5.7	本章小结	137
第 6 章	软件过程的建立与管理	/138
6.1	过程的建立	138
6.1.1	确定过程模型	140
6.1.2	确定活动	141
6.1.3	确定活动间的关系	141
6.1.4	将每个活动的有用信息文档化	142
6.1.5	剪裁过程文档化	142
6.1.6	改进过程文档化	144
6.1.7	过程获得认可并培训员工	146
6.1.8	不断地使用和改进过程	146
6.1.9	定义软件过程的实例	147
6.1.10	项目计划的编制	155
6.2	过程的监控	155
6.2.1	过程变更处理	156
6.2.2	变更实施	157
6.3	过程改进	157
6.3.1	CMM	158
6.3.2	其他评估模型	160
6.4	本章小结	161
参考文献		/162

绪 论

尽管全世界已有数百万计的软件职业人,但软件工程作为合法的工程学科存在只不过是几年前的事情。软件工程过程作为软件工程学科的一个重要组成部分,必将越来越得到人们的重视。本章将首先回顾软件工程学科的发展史,然后介绍软件工程过程的基本概念及其相关标准。

1.1 软件制造是个复杂的过程

软件伴随着第一台电子计算机的问世诞生了,随后以编写软件为职业的人也开始出现。由于受当时计算机硬件的限制,只有很少数的专业人员才能使用计算机,他们多是经过训练的数学家和电子工程师。编写软件的唯一目的是完成大量的科学计算或使某些重复性工作自动化,把人们从烦琐、耗时的科学计算、工业控制和数据处理等重复劳动中解脱出来。

在计算机系统发展的初期,硬件通常用来执行一个单一的程序,而这个程序又是为一个特定的目的而编制的,因此通用性很差。大多数软件是由使用该软件的个人或机构研制的,个人色彩很浓,当然也没有什么系统的方法可以遵循,软件设计是在某个人的头脑中完成的一个隐式的过程。而且,除了源代码往往没有软件说明书等文档。

后来,人们逐渐认识到软件与硬件的差异:

首先,软件更容易更改,且不需要昂贵的生产线进行批量生产。程序一旦被修改,只需要把修改后的程序再装到其他计算机上即可。这种便于修改的特点,使编程人员和组织开始采用一种“编码和修正”的方式来开发软件。

其次,软件不会被用坏。使用硬件模型来度量软件的可靠性是不够的,而“软件维护”是一个与硬件维护大不相同的活动。软件是无形的,维护费用高。因其不确定性使软件开发

很难按计划进行。为了赶工而在软件开发后期加入更多人的做法只会使工期更长。软件一般有更多的状态、模式和测试途径,使之更难以规格化。Winston Royce 在其 1970 年发表的论文中提到:“为了使用一台价值 500 万美元的硬件设备,需要编写一本 30 页的详细设计说明书来为完全控制生产提供足够的细节;但是为价值 500 万美元的软件提供 1500 页的说明,也只能刚刚比较好地对软件生产进行控制。”

最后,软件需求范围快速扩大,远远超出了当时从事软件开发工作的工程师和数学家能力的范围,也造成了软件人才的短缺。这导致大批非常具有创造力,但缺乏工程经验的人由于企业、政府和服务业对软件的需要,而涌入软件开发行业。“编码与修正”的方式非常适合这些人,他们思想开放,更喜欢按照自己的想法去实现工程。到处可见所谓的“牛仔程序员”,为了保证最后期限,他们可以用整夜时间来快速弥补有缺陷的代码,人们通常会把他们当作英雄。但这些被认为是优秀的程序常常很难被别人看懂,通篇充满了程序技巧,导致维护工作更加困难。

软件的数量急剧膨胀,软件需求日趋复杂,维护的难度越来越大,开发成本之高令人吃惊,失败的软件开发项目却屡见不鲜,出现了“软件危机”。

软件危机迫使人们从技术、人员、管理、开发工具等诸多方面系统化地考虑软件生产和维护过程,于是出现了软件工程。软件工程是借鉴传统工程的原则、方法,以提高质量、降低成本为目的的指导计算机软件开发和维护的工程学科。它包括建造软件的过程、方法、工具和质量要求 4 个方面的内容。

1.2 软件产品与软件过程

软件工程的实质是在合理的时间和费用开销内,为用户提供满意的软件产品。编写软件本身是一项智力型劳动。软件从业人员在从无章法地编程到使用结构化方法、面向对象方法以及现在基于模型驱动的开发方法的探索中,始终保持了高度热情。在支持工具方面,从业务建模工具、需求管理工具到集成测试工具等应有尽有。但随着软件规模与复杂度的提升,纵使招募了集先进的软件开发方法、工具于一身的高素质的软件开发人才,如果软件制造过程很弱,也很难保证项目的成功;同样,过分依赖过程也是危险的。简言之,正如 Margaret Davis 评论的过程和产品的两面性:大约每隔 5~10 年,软件界就会重新定义“问题”,将其焦点从产品问题转移到过程问题。为了增加程序的可维护性,出现了结构化程序设计语言(产品),之后就有了结构化的分析与设计方法(过程),有了数据封装(产品),之后是软件开发能力成熟度模型(过程),之后是面向对象方法(产品),之后是计划驱动的软件过程,之后是模型驱动的开发,之后是敏捷过程,再后就是计划与敏捷的平衡(过程)。

软件界关注的焦点在不断移动,就像钟摆一样。当上一次摆动停止后,又有新力作用上

去。这些摆动本身是有害的,因为它们彻底改变了完成工作的方法,使普通的软件开发人员无所适从,更不用说能很好地使用它了。造成这些摆动的原因,不是只注重软件产品而轻视过程,就是认为过程能解决软件开发中的一切问题,孤立地看待软件产品和过程。当然这些摆动也不能解决“问题”,因为它们注定是要失败的,只要产品和过程被视为二分的而不是二元的。

软件制品及其开发过程明显体现出了产品和过程之间的二元性,因此单从过程或产品角度,很难得到或理解整个软件,如它的语境、使用、含义或价值。实际上人们从软件创造的过程中和从最终产品中可得到同样的甚至更多的满足感。

软件工程不是一项高度智能化的过程,必须根据软件专业技术人员的技术水平及其任务的创造需要进行动态调整。所以需要在组织对标准化和一致性的要求与个人对灵活性的要求之间进行权衡。需要考虑的一些因素包括:

- 由于软件项目各不相同,相应的软件过程也要有所差异。
- 在缺乏通用的软件过程时,组织和项目必须定义满足其特定需要的过程。
- 用于给定项目的过程必须考虑人员的经验水平、产品当前的状态和可用的工具与设施。

一个过程定义了为达到某个确定的目标,需要什么人在什么时间以何种方式做何种工作。对软件工程而言,其目标是构建一个新的软件产品或者完善一个已有的软件产品。一个有效的过程为开发高质量的软件提供准则,它获取并提出当前技术条件下可行的最佳实践方案,因此可降低风险并增强预见性。总的效果是要发扬一种共同的构想与文化。

也正是基于这些共同的认识,在软件工程知识体(Software Engineering Body of Knowledge, SWEBOK)中将软件工程分成软件需求、软件设计、软件编码、软件测试、软件维护、软件配置管理、软件工程管理、软件工程过程、软件工程工具与方法以及软件质量十大知识域,软件工程过程是其中之一。

在这十大知识域中,“软件工程过程”知识域反映了近年软件过程技术的成果,即一个产品的开发不是某个固定的过程模型,而是根据产品应用域、开发企业的文化和资产,专门设计一个最优过程,即不仅要设计产品,还要设计过程;不仅要量度产品的质量,还要量度过程。但 SWEBOK 采取较为保守的态度,强调过程的变更和改进过程,而不是设计新过程。这既符合 ISO/IEC 12207 标准,可行性又强。

在 SWEBOK 中,软件工程过程包括两个层次:

第一层与技术和管理相关,实施的活动是软件获取、开发、维护和退役。第二层也称元层(Meta),涉及定义、实现、测量、管理、变更和改进等活动,后者也称为软件过程工程。

本书内容仅涉及第一个层次,即将用户需求转化为软件所需要的一整套软件工程活动,重点讨论软件产品开发与维护中涉及的核心工程活动。

1.3 为什么要研究软件工程过程

做任何事情都有一个过程,软件开发也不例外。但因早期软件开发的主要障碍是技术——只要能做出来就行。因此,20世纪50年代的软件系统的主要目的是解决科学计算问题。20世纪60至70年代,为了使编程更加容易,出现了新的语言;为了使普通的编程者能把精力集中在应用系统开发上,出现了专门的操作系统来屏蔽计算机硬件系统的复杂度;为了使数据处理更加容易,在操作系统提供的文件系统基础上出现了数据库管理系统。但此时开发的软件对某些特定硬件的依赖性较大,独立的软件较少,因此软件开发技术处于主导地位,工程处于被动状态。

尽管1968年软件界已倡导软件工程,但计算机技术普及的主要障碍还是其技术的发展无法满足人们对软件的需要。直到20世纪90年代中期,人们主要的关注点还是技术,也因此使软件技术和开发方法呈现出飞速发展的势头。

(1) 20世纪80年代中期以前,人们主要使用基于结构化编程和瀑布模型的开发方法。

(2) 从20世纪80年代中期以后,结构化过程编程被面向对象编程取代。

(3) 网络技术成熟,基于单主机计算的开发开始向分布式客户/服务器计算的开发方法转移,然而却没有成熟的规范可以借鉴。瀑布模型受到批评以后,螺旋模型通过多次迭代接近实际的特点,而备受推崇。但如何实施管理与进度预算、如何进行风险评估、如何实施里程碑评审等都没有公认规范可以借鉴,可操作性弱,造成使用困难。

(4) 20世纪90年代中期软件过程、软件生存周期及软件评估规范开始成熟,如ISO/IEC 12207。之后计算技术又发生重大变化,进入新的网络计算时代,软件开发的目标是提供Web服务。应用开发以集成服务为主、自行编码为辅,面向对象深化为构件和接口的链接,通信协议成为编程的重点。这种深刻的变化不要说没有成熟的规范和方法学,就连应用模式也还在探索之中。然而正是这种正在探索中的方法和模式却又在某些行业已经投入使用,且显示了巨大的优越性。

(5) 2000年之后,技术发展已经不再成为人们应用计算机的障碍,软件开发关注的焦点也从原来的单系统开发转移到多系统以及系统中的系统,即群集系统的开发。多系统间的无缝衔接问题,遗留系统问题,系统的稳定性、易用性等要求,再加之快速变化的业务需求、多个涉众利益的均衡等使管理协调的工作量变得越来越大,使软件制造过程变得越来越复杂。这些也迫使人们凭借以往的开发经验,在项目启动之初,制定足够详细的软件开发流程和规范,约束项目有关各方,规范所有参与人的行为,即通过规范化的软件过程来组织参与系统建设的各元素,使其按预定的轨道前进,保证项目的成功。

尽管开发组织不否认好的开发过程对软件项目的成功的保障作用,但当受到技术挑战、需求变更、交付时间和成本等方面压力时,首先被忽略、弱化或破坏的还是软件过程。