

21世纪高等学校计算机科学与技术规划教材

软件工程

RUANJIAN GONGCHENG

任胜兵 邢琳 编著



北京邮电大学出版社
www.buptpress.com

21 世纪高等学校计算机科学与技术规划教材

软 件 工 程

任胜兵 邢 琳 编著

北京邮电大学出版社

内 容 简 介

本书以工程化的软件开发过程为主线,结合实际应用实例,系统地介绍了软件工程有关的标准、概念、原理、方法和技术。全书共分10章,主要内容有:软件生存期过程和模型、可行性研究、软件项目计划、软件需求分析、软件结构设计、软件详细设计、软件测试、软件维护、软件质量和保证体系以及软件复用技术。

全书内容新颖,讲述力求实用、深入浅出、循序渐进。每章均附有习题。

本书可作为高等学校计算机科学与技术专业及相关专业本科“软件工程”课程教材,亦可作为从事软件开发的科技人员的参考书、培训教材等。

图书在版编目(CIP)数据

软件工程/任胜兵,邢琳编著. —北京:北京邮电大学出版社,2004

ISBN 7-5635-0844-9

I. 软... II. ①任...②邢... III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字(2004)第 034645 号

书 名: 软件工程

编 著: 任胜兵 邢 琳

策 划: 三文工作室

E-mail: sanwen99@mail.edu.cn

责任编辑: 陈露晓 付晓霞

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(100876)

电话传真: 010-62282185(发行部) 010-62283578(传真)

经 销: 各地新华书店

印 刷: 北京通州皇家印刷厂

开 本: 787mm×960mm 1/16

印 张: 24.25

字 数: 438 千字

版 次: 2004 年 5 月第 1 版 2006 年 1 月第 4 次印刷

ISBN 7-5635-0844-9/TP·108

定 价: 33.50 元

如有质量问题请与发行部联系

版权所有 侵权必究

21 世纪高等学校计算机科学与技术规划教材

编 委 会

主 任	陈火旺	中国工程院院士,国防科技大学教授
委 员	周立柱	清华大学计算机系主任
	杨放春	北京邮电大学计算机科学与技术学院院长
	杨学军	国防科技大学计算机学院院长
	徐晓飞	哈尔滨工业大学计算机科学与技术学院院长
	李仁发	湖南大学计算机与通信学院院长
	卢正鼎	华中科技大学计算机学院院长
	李志蜀	四川大学计算机学院院长
	戴居丰	天津大学信息学院、软件学院院长
	蒋昌俊	同济大学计算机科学与工程系主任
	何炎祥	武汉大学计算机学院院长
	周兴社	西北工业大学计算机系主任
	陈志刚	中南大学信息学院副院长
	姜云飞	中山大学软件学院院长
	周昌乐	厦门大学软件学院院长
	齐 勇	西安交通大学计算机科学与技术系主任
	赵书城	兰州大学计算机学院院长
	孟祥旭	山东大学计算机学院院长

序

自 20 世纪 80 年代以来,高等学校计算机教育发展迅速,计算机教育的内容不断扩展、程度不断加深。特别是近十年来,计算机向高度集成化、网络化和多媒体化发展的速度一日千里;社会信息化不断向纵深发展,各行各业的信息进程不断加速;计算机应用技术与其他专业的教学、科研工作结合更加紧密;各学科与以计算机技术为核心的信息技术的融合,促进了计算机学科的发展,各专业对学生的计算机应用能力也有更高和更加具体的要求。

基于近年来计算机学科的长足发展,以及国家教育部关于计算机基础教学改革的指导思路,我们确立了这套“21 世纪高等学校计算机科学与技术规划教材”的编写计划与编写思想。教材是教学过程中的“一剧之本”,是高校计算机教学的首要问题。该套系列教材编写计划的制定凝聚了编委会和作者的心血,是大家多年来计算机学科教学和研究成果的体现,并得到了陈火旺院士的亲自指导与充分肯定。

这套系列教材由北京邮电大学出版社三文工作室精心策划和组织。编写过程中,充分考虑了计算机学科的发展和《计算机学科教学计划》中内容和模块的调整,使得整套教材具有科学性和实用性。整套系列教材体系结构按课程设置进行划分。每册教材均涵盖了相应课程教学大纲所要求的内容,既具备学科设置的合理性,又符合计算机学科发展的需要。从结构上遵循教学认知规律,基本上能够满足不同层次院校、不同教学计划的要求。

各册教材的作者均为多年来从事教学、研究的专家和学者,他们有丰富的教学实践经验,所编写的教材体系结构严谨、内容充实、层次清晰、概念准确、论证充分、理论联系实际、深入浅出、通俗易懂。

教材建设是一项长期艰巨的系统工程,尤其是计算机科学技术发展迅速、内容更新快,为使教材更新能跟上科学技术的发展,我们将密切关注计算机科学技术的发展新动向,以使我们的教材编写在内容上不断推陈出新、体系上不断完善,以适应高校计算机教学的需要。

21 世纪高等学校计算机科学与技术规划教材编委会

2003 年 12 月

前 言

近年来,软件工程得到学术界和产业界的广泛重视,大量的理论和应用研究成果不断涌现,形成和促进了软件产业的飞速发展,在社会生活中扮演着愈来愈重要的角色。实践表明,不掌握一定的软件工程知识,不按软件工程的有关原理进行软件开发,不积极学习新的软件开发方法和技术,就不能高效高质量地开发软件。本教材正是为了适应软件工程高等教育的需要,由多年从事软件工程教学和科研的教师编写而成的。

本书系统地介绍了软件工程的有关概念、标准、原理、方法和技术,其目标是使学生理解有关软件生存期过程与模型概念和软件工程基本原理,掌握建造软件系统的有关方法、技术和标准。本课程是一门工程性课程,实践性很强,学生在学习时除了对概念、原理等的理解之外,更应结合实际,注重方法、技术等运用。

全书共分 10 章。第 1 章简述软件工程的产生、软件工程学的研究对象与原理、软件开发工具与方法、软件工程的生存期过程和模型以及软件工程标准和文档。第 2 章对系统的组织结构、处理流程和数据分析进行了阐述,详细叙述了成本估计和效益分析的有关内容。第 3 章介绍与项目计划有关的风险分析、进度安排和项目组织的有关内容。第 4 章至第 8 章叙述软件开发和维护阶段的概念、任务、原理、方法和技术。第 9 章介绍软件质量的有关概念、分析技术和质量保证体系等。第 10 章主要介绍有关软件复用的概念和技术,包括领域工程、组件技术、设计模式和应用框架等。如果教学时间有限,则可以只讲授第 1 至 8 章,其内容依然是完整的。为了便于学生的学习,本书在介绍有关内容时,给出了丰富的例子,并且每章均附有相当数量的练习题。

本书主要是在编者近年来讲授软件工程或相关课程的讲义和研究成果的基础上改写而成的,由任胜兵和邢琳共同编著。在改写时,充分考虑了目前高等学校对软件工程课程教与学的要求。一方面强调软件工程的实用性和实时性。对于理论性要求较高而实际中不常用的一些内容,如程序正确性证明、形式化需求分析等,没有详细讨论,而对于当前流行的一些软件 Engineering 内容,如 UML 的应用、复用技术、用例建模、类设计原则等,介绍的比较充分。另一方面在介绍软件工程的有关内容时不仅考虑内容的深度和广度,而且也考虑到教学内容的连续性和系统性。例如编码部分,目前高校均有相当的课程与此有关,故本书没有再对此进行重复介绍,又如本书将所有相关内容纳入到软件生存期的各阶段来介绍,

而不是一个一个方法单独处理。

在编写过程中,中南大学信息科学与工程学院博士生导师陈松乔教授提出了不少指导意见,为本书的编写奠定了基础。此外,中南大学信息科学与工程学院副院长杨路明教授、王国军博士等为本书的编写提供了许多很有价值的帮助。谨在此向他们表示衷心感谢。

由于作者水平所限,加之时间仓促,书中错误和不当之处在所难免,还望各位读者批评指正。

编 者

2004 年 3 月

目 录

第 1 章 绪论 (1)	第 4 章 需求分析 (76)
1.1 软件及其发展 (1)	4.1 需求分析的特点 (76)
1.2 软件工程 (5)	4.2 需求收集 (77)
1.3 软件生存期过程 (7)	4.3 数据流建模..... (80)
1.4 常用软件生存期模型 (10)	4.4 IDEF0 功能建模 (89)
1.5 软件工程标准 (15)	4.5 IDEF1X 数据建模 (95)
1.6 软件开发方法 (19)	4.6 UML 建模语言 (101)
1.7 软件工程工具和环境 (22)	4.7 用例建模 (111)
1.8 软件文档 (26)	4.8 对象建模 (118)
1.9 小结 (30)	4.9 需求规格说明与验证 (132)
习题 1 (30)	4.10 小结 (137)
第 2 章 可行性研究 (32)	习题 4 (137)
2.1 可行性研究内容与步骤 ... (32)	第 5 章 概要设计 (140)
2.2 系统分析 (35)	5.1 软件设计基本原理 (140)
2.3 成本估计 (41)	5.2 面向数据流图的设计 (147)
2.4 效益分析 (48)	5.3 面向 IDEF0 图的设计 ... (157)
2.5 可行性研究文档 (51)	5.4 类设计原则 (159)
2.6 小结 (54)	5.5 面向对象设计 (166)
习题 2 (55)	5.6 概要设计文档及其复审... (175)
第 3 章 软件项目计划 (56)	5.7 小结 (177)
3.1 风险分析 (56)	习题 5 (178)
3.2 进度安排 (61)	第 6 章 详细设计 (180)
3.3 项目组织 (66)	6.1 详细设计的目标与任务 ... (180)
3.4 软件项目开发计划文档 ... (72)	6.2 详细设计图形描述工具 ... (181)
3.5 小结 (74)	6.3 Jackson 程序设计方法 ... (188)
习题 3 (74)	6.4 Warnier 程序设计方法 ... (193)
	6.5 面向对象详细设计 (199)

6.6 程序规格说明文档及复审	(207)	9.2 软件复杂性分析	(266)
6.7 小结	(207)	9.3 软件可靠性分析	(276)
习题 6	(208)	9.4 ISO 9000 软件质量体系 ...	(284)
第 7 章 软件测试	(209)	9.5 软件配置管理	(290)
7.1 软件测试基础	(209)	9.6 软件过程能力成熟度模型简介	(294)
7.2 代码复审	(214)	9.7 小结	(301)
7.3 白盒测试	(216)	习题 9	(302)
7.4 黑盒测试	(222)	第 10 章 软件复用技术	(305)
7.5 单元测试	(230)	10.1 软件复用概述	(305)
7.6 集成测试	(234)	10.2 领域工程	(310)
7.7 确认测试	(238)	10.3 组件技术概述	(317)
7.8 系统测试	(240)	10.4 公共对象请求代理体系结构	(322)
7.9 调试	(242)	10.5 组件对象模型	(327)
7.10 小结	(244)	10.6 EJB 模型	(332)
习题 7	(244)	10.7 可复用面向对象设计模式	(335)
第 8 章 软件维护	(246)	10.8 应用框架	(355)
8.1 软件维护的基本概念	(246)	10.9 小结	(361)
8.2 软件维护过程	(251)	习题 10	(361)
8.3 软件可维护性	(255)	附录 A	(363)
8.4 软件再工程技术	(257)	附录 B	(371)
8.5 小结	(260)	参考文献	(378)
习题 8	(260)		
第 9 章 软件质量及保证体系	(261)		
9.1 软件质量	(261)		

第1章 绪 论

在人类认识和征服自然的过程中,对于工作量大、涉及人员多的工作,往往需事先制订计划,然后按计划进行,这便是工程。秦昭襄王 51 年(公元前 256 年),李冰在其子二郎的协助下,邀集有治水经验的当地人民,对岷水东流的地形和水情作了实地考察,通过实施引水、分水和排砂,消除了水患,使川西平原变成“天府之国”。二千二百多年来,都江堰水利工程至今仍然连续使用,仍发挥着巨大的效益。就软件而言,随着计算机应用的深入,其开发和维护的工作量愈来愈大、涉及的人员愈来愈多。同水利工程有计划地实施水利开发活动一样,软件工程也通过实施一系列有计划的软件开发和维护活动来进行,其目的在于以较少的代价来获取高质量的软件。

本章主要使读者了解软件工程的概貌,其内容包括:软件工程学科的形成,软件工程的研究对象,软件生存期过程与模型,软件工程标准、方法、工具和环境。

1.1 软件及其发展

计算机系统由软件和硬件这两个密不可分的部分组成。硬件是系统中看得见、摸得着的物理设备。硬件的发展从电子管到晶体管,再到集成电路,直至超大规模、超高速集成电路,性能越来越强,价格越来越便宜。软件是系统中看不见、摸不着的逻辑部分,以程序、数据和文档的形式出现。程序是能够完成预定功能和性能的可执行的指令序列;数据是使程序能够适当地操纵信息的数据结构;文档描述了与程序开发、使用和维护有关的文字或图形资料。硬件性能的提高,极大地促进了更为复杂的软件系统的出现。软件系统的复杂性给软件系统的开发和维护带来一系列问题,由此也推动软件不断向前发展。

1.1.1 软件特征

了解软件的特征并据此明白软件与硬件之间的区别对更好地理解软件的发

展是必要的。同硬件相同,软件是一种无形的逻辑实体,由此具有与硬件不同的特殊性质。

1. 软件不是传统意义上的“制造”产生的,而是“研发”出来的

软件制造只是简单的拷贝,一旦软件项目研发成功,通过复制就可以产生大量的软件产品,因而软件的成本和质量主要取决于软件的研发。硬件的成本和质量则主要取决于硬件研制成功后的重复制造过程。虽然“制造”和“研发”都依赖于人,但参与者和完成的工作之间的关系不同。同时,尽管“制造”和“研发”的目标是构造“产品”,但两者采用的方法不同。这意味着,软件项目的管理、软件产品的保护等等必须不同于硬件。

2. 软件不会被“用坏”

软件和硬件一样,在使用的早期,会由于设计或研发、制造的错误或缺陷造成故障,但随时间的推移,由于错误和缺陷的改正,故障率逐步降低。在使用的后期,硬件会因为磨损和老化等问题,故障率逐步上升。当故障率达到一定程度后,硬件就会因“用坏”而寿终,软件不会在使用过程中出现磨损和老化,因此软件不会因“用坏”而寿终。但软件会因环境或需求的变化必须多次修改(维护)。在实际中,软件的每次维护都不可避免地引入新的错误。同时,软件的维护过程实质是一次研发过程,不能仅通过软件的重复制造来解决。而硬件因磨损或老化等导致的“用坏”则可以通过重复制造的备用零件替换来解决。所以,软件的维护要远比硬件的维护复杂得多。

3. 软件大多是“定制”的

现代工业社会中,硬件基本都通过组装不同生产厂商的各种零部件而设计和制造出新产品,定制的成分比较少。软件至今很少能做到利用现成的部件通过组装形成所需的软件。软件大多是为用户专门“定制”而成的。对于软件开发人员而言,软件开发工作依然是一种高强度的脑力劳动。软件复用技术、自动生成技术等,在现有的软件项中使用的比率仍然很低,软件产品开发还未完全摆脱传统的手工开发方式。正因如此,软件开发的质量和效率受到很大的限制。

4. 软件成本难于估计

硬件成本主要取决于研发后的制造过程,容易进行成本估算并制订周密计划。软件成本主要取决于软件的研发过程,一方面,软件需求和程序逻辑结构复杂,且其复杂性与软件技术的发展不相适应,同时软件技术发展落后的差距越来越大;另一方面,软件产品的研发需要大量的脑力劳动。因此,软件成本难于估计且越来越昂贵。人们对软件成本的不准确估计往往造成软件项目计划失效,进而直接影响软件项目的成败。

1.1.2 软件分类

软件的分类从另一方面反映了软件的发展。前面从软件与硬件的比较中得出了各种软件的共同特点。但不同类型的软件,其开发和维护又有各自的不同特点。由于软件本身的复杂性,难以找到一个统一的严格分类标准。本节从各种不同的角度来对软件进行分类。

1. 按应用功能分类

- 系统软件:与计算机系统硬件紧密交互,协调计算机系统各部分工作的软件。例如操作系统、设备驱动程序及通信处理程序等。系统软件是计算机系统必不可少的一个组成部分。

- 支持软件:协助使用者开发软件的工具性软件。例如程序编译器、自动化测试软件、系统分析辅助工具及软件开发管理工具等。

- 应用软件:为使一个计算机系统得到某种功能而专门开发的软件。例如:商业信息处理软件、工程和科学计算软件、人工智能软件及智能产品嵌入软件(如微波炉的按钮控制、汽车中的燃料控制等)、多媒体播放软件等。有时,支持软件和应用软件的划分边界比较模糊,如字处理软件,既是支持软件辅助软件开发,又可看成应用软件。

GB/T 13702—92 按计算机软件的基本特征和主要功能,采用线性分类法对软件进行了分类。对于每一种类型,给出了对应的代码,其表示形式如图 1-1 所示。

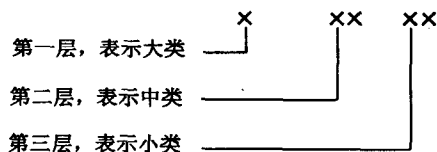


图 1-1

例如,系统软件、支持软件和应用软件是三大类,其编号分别为 1、3、6。操作系统软件的编号为 110,表示其属于系统软件大类;软件开发工具的编号为 310,表示其属于支持软件大类;测试工具的编号为 31040 表示其属于软件开发工具中类。

2. 按服务对象的范围分类

- 项目软件:软件开发机构受特定用户委托开发而成的软件。例如,电信管理系统、空中交通管制系统、军用防空指挥系统、生产过程控制系统等。一般情况下,项目软件在合同的约束下开发。为了争取软件开发合同,软件开发机构必须重视质量管理,而软件开发机构的技术实力、开发经验以及社会信誉等也相当重要。

- 产品软件:软件开发机构直接为市场开发的软件。例如,字处理软件、多媒体播放软件、游戏软件、教育软件等等。产品软件的功能、性能、价格和售后服务对开发机构参与市场竞争有重要影响。

1.1.3 软件发展与软件危机

软件发展大致可以划分为三个阶段:程序设计阶段、程序系统阶段和软件工程阶段。

1. 程序设计阶段

20 世纪 50 至 60 年代,人们使用机器语言或汇编语言书写程序。由于硬件价格昂贵、存储容量小、运行可靠性差等,人们在编写程序时追求编程技巧、追求程序运行效率,程序设计被看作是一种发挥个人创造才能的技术领域。而且,当时人们认为写出的程序只要能在计算机上运行并得出正确的结果就行,程序的写法可以不受任何限制。这些氛围,使得编写的程序难于理解、修改困难。在程序设计阶段,软件的生产主要是专家的个体手工劳动。

2. 程序系统阶段

在 20 世纪 60 至 70 年代,硬件在性能和价格方面有了明显的改善,计算机的应用领域不断扩大,软件的需求也由此不断增长。软件由于处理领域的扩大而变得复杂,软件的生产难于由单个个体完成,于是形成了“软件作坊”,使用高级程序设计语言来开发程序。在程序系统阶段,开发人员开始注重程序设计风格,提出了结构化程序设计方法与模块化设计思想。但“软件作坊”基本上还是沿用程序设计阶段形成的个体手工开发方式。在此阶段,不仅大量的软件需要开发和维护;同时,已有的软件开发和维护技术以及组织管理不适应大规模复杂软件的开发。这样,软件开发和维护便出现一系列严重问题,即软件危机。这些问题不仅仅限于所开发出的软件成本高且质量得不到保证,还包括如何开发软件,如何维护现有软件等等。与经济危机“商品供过于求”的特点相比,软件危机则是“软件供不应求”。究其原因,一方面是由软件本身的特点引起的,如软件越来越复杂、软件故障难于检测等;另一方面则是因为软件开发和维护以及组织管理不当造成的,如软件开发的个性色彩太浓、软件需求不充分、软件开发过程无序等。

3. 软件工程阶段

1968 年北大西洋公约组织(NATO)的科学委员会在联邦德国召开的有关软件危机的讨论会上,首次正式提出了“软件工程(Software Engineering)”的概念,其主要思路是:要把人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法,特别是人类从事计算机硬件研究和开发的经验教训,应用到软件的开发和维护中来。从 1968 年到现在,软件开发有了很大的进步:软件开发工具和开发环境不断改进;软件工程标准和规范不断完善;面向对象技术、软件质量保证、软件复用技术等逐渐得到人们的认同等等。但到目前为止,软件

危机并未完全摆脱,还需人们对软件工程进行更长久更深入的研究。

1.2 软件工程

软件工程是指导软件开发和维护的工程性学科,它以计算机科学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,把经过时间证明是正确的管理措施和当前能够得到的最好的技术、方法相结合,以期用较少的代价获取高质量的软件。

1.2.1 软件工程的研究内容

软件工程是一门综合性的交叉学科,主要涉及计算机科学、管理科学和数学等。计算机科学侧重于理论研究,其成果可应用于软件工程,而软件工程则强调如何有效地建造一个软件系统。管理科学中的方法和原理可用来进行软件生产的管理,包括费用估算、制订计划等。数学可用来建立软件开发中的各种算法和模型。

软件工程作为一门独立的学科,其主要的研究内容是与软件开发和维护有关的4个方面的内容:标准和规范、过程和模型、方法和技术、工具和环境,软件工程管理则贯穿于这4个方面。

作为工程而言,标准化、规范化可以使各种工作有章可循,进而提高软件的生产效率和软件产品的质量。软件工程标准主要有5个层次:国际标准、国家标准、行业标准、企业规范和项目规范,其详细内容将在后面继续讨论。

所有软件开发和维护都由一系列过程所构成,建立在一定的标准和规范的基础上,将方法和技术、工具和环境相结合,以便合理、及时地进行软件开发和维护。软件生存期模型则将软件过程有机结合起来,提供一个结构框架,明确主要活动和任务,忽略次要细节,以利于软件开发和维护的各类人员理解,适应不同的项目,有时也称为过程模型,如瀑布模型、原型模型等。由此看来,过程强调的是具体的活动和任务,模型则突出表现过程的有机结合。

软件开发和维护方法体现了软件开发和维护人员看待系统的立场和观点,即方法论意义上的方法。例如,结构化方法认为系统是由一些结构化的功能相互联系、相互作用而构成;面向对象方法则认为系统是由一些对象的相互联系、相互作用而构成。技术则是方法的具体实现,由若干步骤组成,突出“如何做”,有时也不加区别地称之为方法,即技术方法,如SA/SD(Structured Analysis/Structured Design)方法是一种结构化分析与设计技术、OMT(Object Modeling

Technique)方法是一种面对象分析和设计技术等等,读者可以依据有关的上下文来加以区分。除此之外,还有一些与软件开发和维护有关的辅助技术,如软件复杂性分析技术、软件可靠性分析技术等。

工具为软件开发和维护的方法和技术提供了自动或半自动的软件支持,以提高软件生产质量和效率,降低软件开发和维护成本,如编译程序、测试工具等。将各种工具结合起来,连同有关的软硬件便形成软件开发和维护环境,其目的是使工具支持整个软件生存周期,如北大的青鸟系统、中科院软件所的XYZ系统等。

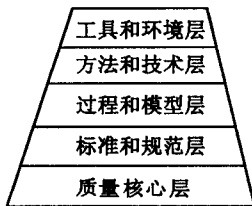


图 1-2 软件工程层次结构

软件工程的四个方面的研究内容构成了以软件质量为核心的层次结构,如图 1-2 所示。全面质量管理和类似的管理技术孕育了不断发展和提高的企业文化,而正是这种企业文化导致了不断成熟的软件工程标准和规范、过程和模型、方法和技术、工具和环境的发展。

1.2.2 软件工程的基本原理

1983 年,美国著名的软件工程专家 Boehm 综合 100 多条关于软件工程的准则和信条,并总结 TRW 公司多年的软件开发经验,提出了软件工程的 7 条基本原理。Boehm 认为这 7 条原理是确保软件产品质量和开发效率的原理的最小集合。

1. 用分阶段的生命周期计划严格管理

统计表明,在不成功的软件项目中有一半左右是由于软件开发和维护计划不周造成的。在软件开发和维护的生命周期中,需要完成许多性质不同的工作。把软件生命周期划分为若干阶段,并相应地制订出切实可行的计划,然后严格按照计划对软件的开发和维护工作进行管理,是软件项目成功的关键。

Boehm 认为,在软件的整个生命周期中应该制订六类计划,即:项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划和运行维护计划。

2. 坚持进行阶段评审

统计表明,大部分的软件错误是在编码之前造成的,约占 63%。同时,错误发现得越晚,改正错误付出的代价也就越大。因此,必须坚持严格的阶段评审,以便尽早发现错误。

3. 实行严格的产品控制

由于外部环境的变化,用户需求的改动往往是不可避免的。但在软件开发过程中改变需求需要付出较高的代价,因此不能随意改变需求,要采用严格的产

品控制技术,其中主要是实行基准配置管理。所谓基准配置又称基线配置,它们是经过阶段评审后的软件配置成分,如各阶段产生的文档或程序代码。基准配置管理也称变动控制:一切有关软件修改的建议,特别是涉及到对基准配置修改的建议,必须经过严格的规程进行评审,获得批准后才能进行有关修改。

4. 采纳现代程序设计技术

从提出软件工程概念开始,人们都致力于研究各种新的程序设计技术。20世纪60年代末提出的结构程序设计技术,已经成为当时绝大多数人公认的先进的程序设计技术。以后又进一步发展出各种结构化分析和结构化设计技术。实践表明,采用先进的技术既可提高软件开发的效率,又可提高软件维护的效率。

5. 结果应能清楚地审查

软件是一种看不见、摸不着的逻辑产品,从而使软件产品的开发过程比一般产品的开发过程难于评价和管理。因此,应该根据软件项目的总目标及完成期限,规定开发组织的责任和产品质量标准,提高软件开发过程的可见性,从而使所得的结果能够清楚地审查。

6. 开发小组的人员应少而精

当开发小组人员数为 N 时,可能的通信路径有 $N * (N-1)/2$ 条。显然,随着人数 N 的增大,人员之间的通信路径急剧增加,造成通信开销的剧增。同时,高素质的人员开发软件的效率比低素质人员开发软件的效率可能高出几倍甚至几十倍,而且错误数明显比低素质人员开发的软件中的错误数少。因此,软件开发小组的人员组成应该少而精。

7. 承认不断改进软件工程实践的必要性

Boehm认为:遵循软件工程的前6条原理,就能够较好地实现软件的工程化生产,但不能保证跟得上技术的不断进步。承认不断改进软件工程实践的必要性,不仅要积极主动采纳新的软件技术,而且应不断总结经验,收集软件开发和维护过程中的有关数据,例如:进度和资源耗费数据、出错类型和统计数据等。经验和数据不仅可用于评价新的技术,而且为新技术的发展提供了重要依据。

1.3 软件生存期过程

软件生存期过程规定了获取、供应、开发、操作和维护软件时,要实施的过程、活动和任务。其目的是为各种人员提供一个公共的框架,以便可以使用“相同的语言”在自己的环境中创作和管理软件。1995年制订和公布的国家标准

《GB/T8566—1995 信息技术——软件生存期过程》定义了软件生存期 7 个主要过程：管理过程、获取过程、供应过程、开发过程、操作过程、维护过程和支持过程，它们之间的作用和关系如图 1-3 所示。

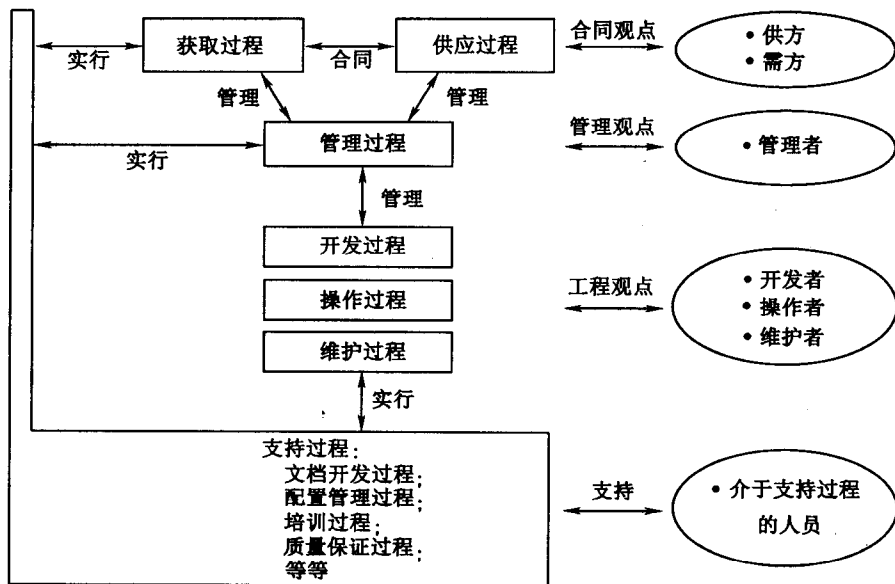


图 1-3 软件生存期过程——作用和关系

1.3.1 获取和供应过程

获取和供应过程从合同的观点分别定义了需方和供方的活动和任务。获取过程从定义软件产品或服务的获取需求开始，接着准备并公布标书、选择供方和管理获取过程，直到系统验收。供应过程的开始方法有两种：一是决定准备一项建议以应答需方的标书；二是就提供一个含有软件的系统（或系统的一个部件、一个产品或一项服务）与需方签订合同或协议。接着就是规定为了管理和保证项目所需的步骤和资源，其中包括制订项目计划和实施计划，直至向需方交付系统、产品或服务。

1.3.2 管理过程

管理过程从管理的观点上定义了一个机构在管理自己的过程中管理者所需的活动和任务。管理过程从对将要采取的过程规定需求开始。接着，管理人员应当制订实施此过程的计划并实施该计划；监督过程的实施；调查、分析和解决执行此过程中发现的问题。最后，评审和评价在过程实施中所完成的产品，保证