21世纪 高等学校本科系列教材

计算机科学与技术专业

Ruanjian Gongcheng

软件工程

主 编 彭 龔

重庆大学出版社
http://www.cqup.com.cn

内容简介

本书主要以工程化的软件开发方法为主导，系统、全面地介绍这门课程的原理、方法及应用。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。全书共分9章，内容包括软件工程概论、软件计划与可行性研究、软件需求分析、软件结构设计、软件详细设计、软件编码设计、软件测试、软件维护和面向对象的软件开发技术概述。

本书可作为高等院校“软件工程”课程的教材或教学参考书，也可作为从事计算机工作的科技人员学习软件工程的参考书。

图书在版编目(CIP)数据

软件工程/彭奘主编. —重庆：重庆大学出版社，2011.2
计算机科学与技术专业本科系列教材
ISBN 978-7-5624-5950-7

Ⅰ.①软… Ⅱ.①彭… Ⅲ.①软件工程—高等学校—教材 Ⅳ.①TP311.5

中国版本图书馆CIP数据核字(2011)第014178号

软 件 工 程

主编 彭 奘
策划编辑：曾显跃
责任编辑：文 鹏 陈 力 版式设计：曾显跃
责任校对：秦巴达 责任印制：赵 晟

*

重庆大学出版社出版发行
出版人：邓晓益
社址：重庆市沙坪坝正街174号重庆大学(A区)内
邮编：400030
电话：(023) 65102378 65105781
传真：(023) 65103686 65105565
网址：http://www.cqup.com.cn
邮箱：fxk@cqup.com.cn (营销中心)
全国新华书店经销
重庆升光电力印务有限公司印刷

*

开本：787×1092 1/16 印张：16 字数：399千
2011年2月第1版 2011年2月第1次印刷
印数：1—3 000
ISBN 978-7-5624-5950-7 定价：29.00元

前言

软件工程的概念是在1968年NATO举行的计算机科学大会上为应对“软件危机”而提出的。从此软件工程这一词汇很快就频繁出现在计算机刊物和会议上，在大学逐渐形成了课程，内容不断丰富、完善。IEEE-CS/ACM联合工作组1991年制定的计算教程CC1991将“软件工程”列为计算学科的九个知识领域之一，确立了软件工程课程的地位；2001年制定的计算教程CC2001进一步将“软件工程”列为计算学科下的五个二级学科之一，并作为相关学科的一个知识领域，于2004年发布了软件工程知识体SWEBOK、软件工程教育知识体SEEK的正式版本，确定了软件工程学科和课程的基本内容和教学的基本要求。我国也认识到软件工程作为一个学科方向的重要性，教育部于2006年单独成立了计算机科学技术教学指导委员会软件工程分委员会，并颁布“软件工程（本科）专业规范”。

软件工程是一门迅速发展的新兴学科，现在软件工程已成为计算机科学技术的一个重要的分支，一个异常活跃的领域，是继程序设计课之后，对提高学生软件开发能力有重要作用的一门课程。它研究范围非常广泛，包括技术方法、工具和管理等许多方面，即通过改进程序结构，革新软件生产工具和生产方式，以及关于软件可靠性技术的研究，更有效、更经济地开发软件产品，有力地促进软件生产的工程化和软件产品的商品化。

本书着重从实用的角度讲述软件工程的基本原理、概念和技术方法，同时也注意该书的全面性、系统性。力求做到理论与应用相结合，并注意于软件技术中的其他课程衔接。通过该门课程的学习，旨在帮助学生在掌握软件工程基本概念、技术方法的基础上，训练学生的软件分析思维能力，提高软件设计水平，为今后的软件开发打下坚实的基础。

本书共分为9章。第1章，软件工程概论，主要介绍软件和软件工程的相关概念，以及软件工程的基本原理和方法；第2章，软件计划与可行性研究，主要介绍可行性分析，成本/效益分析技术和方法，以及如何制订科学的软件开发计划；第3章，软件需求分析，主要介绍软件需求分析的任务、过程、方法和工具；第4章，软件结构设计，主要介绍软件结构设计的概

念、原理、方法及具体运用过程；第5章，软件详细设计，主要介绍结构化程序设计的思想及描述工具，面向数据结构的详细设计方法；第6章，软件编码设计，主要介绍语言的特性、选择的原则以及编码的风格和效率；第7章，软件测试，主要介绍软件测试概念、方法、过程以及用例设计；第8章，软件维护，主要介绍软件维护的有关概念、方法；第9章，面向对象的软件开发技术概述，主要介绍面向对象的概念、对象分析、对象设计及对象实现的方法及其应用，以及组件技术。

本书第1章、第2章、第3章、第5章、第8章和第9章由彭奘编写，第4章由何华平编写，第6章由魏扬编写，第7章由陈光建编写。全书由彭奘统稿，有关的老师和同事参与了大纲的讨论，在此表示感谢。

限于编者的学识和经验，书中难免有错误和不当之处，恳请读者批评指正。

编　者

2010年10月

目 录

第 1 章
软件工程概论

随着计算机系统技术的发展和计算机应用需求的日益扩大，计算机软件的开发、维护工作显得越来越重要。如何才能开发出用户满意的软件；如何以较低的成本开发出高质量的软件；怎样使所开发的软件在运行过程中容易维护，从而延长软件的使用期限；如何提高软件开发、维护过程中的自动化程度、提高软件开发效率；软件工程如何管理等，这些就是软件工程研究的问题。软件工程的任务就是要在软件的开发策略及设计方法上引入新秩序，建立软件工程新规范，目的是在规定的时间、规定的开发费用内，开发出满足用户需要的、质量合格的软件产品。

1.1 软件综述

随着计算机系统的发展，计算机的应用日趋广泛，计算机软件自身在开发技术及编程方法上存在严重的滞后和无序，导致软件的供给、质量与巨大的社会需求之间存在差异的“软件危机”困扰，为了更好地开发和维护软件，软件工作者在20世纪60年代后期开始认真研究消除软件危机的新途径，从而逐渐形成了一门新兴的工程学科——计算机软件工程学（简称为软件工程）。

1.1.1 软件的发展

自从1946年电子计算机发明以来，计算机软件随着计算机硬件的发展而逐步发展，计算机软件和硬件一起构成了计算机系统。一开始只有程序的概念，后来出现了软件的概念。软件这个概念，从它出现之时，就带有一层神秘的色彩，对于人类而言是一个全新的东西。人们对软件的认识经历了一个由浅到深的过程。随着计算机硬件性能的极大提高和计算机体系结构的不断变化，计算机软件系统日趋成熟和更为复杂，从而促使计算机软件的角色发生了巨大的变化，其发展历史大致可以分为4个阶段。

（1）程序设计阶段

20世纪40年代中期到20世纪60年代初期，软件是作为硬件的附属物而存在的。硬件制造商按用户需要设计制造硬件系统，同时还必须为其设计专用的执行软件。在这个阶段，软

件实际上只是几种程序设计语言编制的程序，计算机程序用来描述计算任务的处理对象和处理规则，而这些程序又只能在特定的计算机上运行，软件的通用性较差，程序的设计与编制过程仍停留在个人凭感觉驾驭的技艺阶段，除了程序清单外，没有其他任何文档资料。

(2)“程序 + 文档”的软件阶段

20 世纪 60—70 年代，硬件技术的发展大大改善了计算机的通用性，这就为软件的发展创造了有利条件。在这个时期，软件开始作为独立的产品在市场上出售，出现了软件产品和“软件作坊”的概念，设计人员开发软件不再像早期阶段那样只为自己的研究工作需要，而是为了用户更好地使用计算机，但“软件作坊”仍然沿用早期形成的个体式的软件开发方法。随着计算机应用的日益普及，软件需求量急剧膨胀，软件业出现了开发热潮。但软件作为“思维产物”的本性，使得软件的开发、运行及维护极其困难，在程序运行时发现的错误必须设法更正；用户有了新需求时，必须相应地修改或添加程序；硬件或操作系统更新时，又可能要修改程序以适应新的环境。这样，软件的维护工作以惊人的比例耗费着资源。更严重的是，程序设计的个体化和作坊化特性造成了软件最终是不可维护的，软件开发成功的可能性及软件的质量已不能够完全为设计人员的意愿所控制，从而出现了早期的软件危机。危机的产生开始从深层次上揭示了软件在开发策略及设计方法上存在的严重缺陷，以及软件在开发的过程管理中缺乏有效的监督审查手段。人们随之也就开始寻求采用软件工程的方法来解决软件危机问题。

(3)软件工程阶段

20 世纪 70 年代到 80 年代末期。软件危机的发生，促使人们冷静下来认真反思软件开发过程中存在的一系列问题。硬件技术的发展，使得硬件的功能越来越强、性能越来越高、成本越来越低、可靠性越来越好、自动化程度越来越高。而软件的开发仍然停留在个人技艺、手工作坊式的水平上，缺少一整套工程化的、可以看得见设计进展、能对软件质量及进度进行追踪控制的规范化方法。软件在分析设计方法上的滞后，使得软件跟不上硬件的发展速度。1968 年在北大西洋公约组织(NATO)的一次专门讨论软件开发的国际会议上，讨论软件危机问题，正式提出“软件工程”(software engineering)的概念，明确了以工程化的原理和方法指导软件的开发、维护，并计划建立一整套工程化的软件开发规范，从此一门新的工程学科诞生了。

软件工程的建立大大促进了软件行业的发展。围绕着软件工程的目标和内容，以及与之相关的理论、技术及方法相继建立。软件开发过程的规范化、工程化为软件的产业化奠定了坚实的基础。

(4)第四代技术阶段

计算机系统发展不再是单台的计算机和计算机程序，而是面向计算机和软件的综合影响。复杂的操作系统控制的强大的桌面系统，连接局域网和因特网，高带宽的数字通信与先进的应用软件相互配合，产生了综合的效果。计算机体系结构也从主机环境转变为分布式的客户机/服务器环境。

软件开发的第四代技术有了新的发展：计算机辅助软件工程(Computer Aided Software Engineering，CASE)将工具和代码生成器结合起来，为许多软件系统提供了可靠的解决方案；面向对象技术已在许多领域迅速取代了传统的软件开发方法；专家系统和人工智能软件终于走出实验室进入了实际应用；人工神经网络软件展示了信息处理的美好前景；并行计算、网络计算、虚拟现实技术、多媒体技术和现代通信技术使人们开始采用与原来完全不同的方法进行工作。

此外，光计算机、化学计算机、生物计算机和量子计算机等新一代计算机的研制发展，必将给软件工程技术带来一场新的革命。

1.1.2　软件的定义

软件因为其高度的抽象性使人们无法从物理实体上感知它、认识它。随着技术的发展，人们对软件的认识也在加深。从软件工程的观点看，软件（software）是指计算机系统中与硬件（hardware）相互依存的另一部分，包括程序（program）、相关数据（data）及其说明文档（document）。用简洁的公式可表示为：

$$\text{软件} = \text{程序} + \text{数据} + \text{系列文档}$$

其中程序是指按照事先设计的功能和性能要求执行的指令序列；数据是指令操作的对象，是程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的各种图文资料。

软件文档指用自然语言或者形式化语言所编写的文字资料和图表，以人们可读的形式出现的技术数据和信息，用来描述程序的内容、组成、设计、功能规格、开发情况、测试结果及使用方法，如程序设计说明书、流程图和用户手册等。

1.1.3　软件的特性

软件同传统的工业产品相比，由于其本身具有的特殊性质，因此，软件产品具有以下特性：

①软件是信息产品。软件是知识的结晶，具有高度的抽象性和严密的逻辑性，是一种知识、技术、资金高密度的逻辑产品。是用文字、符号表达的以程序和文档的形式出现的智力产品。软件产品是看不见摸不着的，因而具有无形性，是脑力劳动的结晶，保存在计算机存储器和外部存储介质上，通过计算机的执行才能体现其功能和作用。人们要认识、理解或编制软件，只能通过建立相应的符号体系进行。程序设计语言就是人们为了表达软件而设计的符号体系。

②软件的生产过程与硬件不同。计划、需求分析、设计、编程、测试、运行和维护构成了软件的“生产”过程。软件产品提供的是一种全面的解决方案，与传统的工业制品不同，软件的生产过程不能割裂。软件一旦研制开发成功后，通过复制就会产生大量软件产品。

③软件的成本构成和传统工业产品不同。软件的研制主要是进行脑力劳动，软件的成本相当昂贵，风险也大。软件产品的成本主要体现在软件的开发和研制上，而生产（copy）成本却很低。软件产品不需要货架，无库存成本，可以通过因特网销售。因此与传统工业产品相比，软件的销售成本偏低。而伴随软件向服务方向的转移，服务成本在总的产品成本中占的比重越来越高。

④软件没有磨损、老化的问题，只有过时与失效。软件是用符号表达的逻辑产品，它不具备任何物理特征，在使用过程中不会因为磨损而老化。软件的过时是指软件已不能满足用户的要求，必须用新的软件替换。失效是指软件信息被破坏（信息被修改、删除）而无法在计算机中运行。软件产品的这一特性启示，不断地适应用硬件、环境以及需求的变化而对软件进行完善、扩充，促使软件升级，以免软件过早地过时而被市场所淘汰。

⑤软件的开发和运行常常受到不同的计算机软件和硬件平台的限制。软件对硬件和软件环境有着不同程度的依赖性。因此，可移植性、互操作性和跨平台性能是衡量软件质量的重要指标。业界公认的一个成功的软件产品要能够满足用户的需求，按时交付用户使用，并消费最

少的生产成本;在保证可用性的基础上,软件产品还应该具有正确性、一致性、稳定性、可扩充性、可移植性和兼容性等良好的性能。

⑥软件具有可剪裁、可扩展、便于分解和组合以及插入、删除的特性。软件是符号表达的逻辑实体,人们当然就可以按需要进行剪裁、增加、修改、删除软件的各个部分。这一特性为软件的组织和构造提供了一条可行的途径。目前广泛流行的是面向对象方法以及即插即用的"软件构件"生成方法,就是利用了软件产品的这一特性。

⑦软件具有强烈的个人色彩,体现了开发者的个人风格,软件工作牵涉很多社会因素。软件是大脑活动的产物。由于不同的人对问题的思维活动不尽相同,因此其结果——软件也必然不尽相同。许多软件的开发和运行涉及机构、体制和管理方式等问题,软件产品体现着设计人员的思维特点和习惯做法(称为软件的属人性)。这些人的因素,常常成为软件开发的困难所在,直接影响到项目的成败。软件工程中建议采用的工程规范方法的目的,就是要降低软件的属人性,增强软件的可读性和可维护性。

⑧软件开发是一个复杂的过程。软件是人类有史以来生产的复杂度最高的工业产品。软件涉及人类社会的各行各业、方方面面,软件开发常常涉及其他领域的专门知识,这对软件工程师提出了很高的要求。一个庞大的系统软件可能有几千万条指令、由数万人联合开发而成,而软件的开发过程对软件工程管理的要求也越来越高。因此,引入标准化开发和管理的技术手段对软件工程的顺利实施意义重大。

1.1.4 软件的分类

学术界和产业界对于软件并没有严格的分类标准,从软件功能、软件规模、软件工作方式、软件服务对象以及软件市场等不同角度,可以将软件划分为不同的种类。

(1)按软件功能进行分类

①系统软件。指能与计算机硬件紧密配合在一起,使计算机系统各个部件、相关的程序和数据协调、高效工作的软件,如操作系统、数据库管理软件、设备驱动程序以及通讯处理程序。

②支撑软件。指协助用户开发软件的工具性软件和中间件,包括程序人员开发软件产品的工具和帮助管理人员控制开发进程的工具。

③应用软件。指在特定领域内开发,为特定目的服务的一类软件,涉及面最宽,如财务管理系统软件。

(2)按软件所处的层次进行分类

根据软件产品在计算机系统中所处的层次和作用,将软件分为平台软件、中间软件和应用软件。

①平台软件。平台软件指直接控制和协调计算机、通讯设备及其他外部设备,使之发生作用并方便用户使用的软件以及提供中间件支持和运行环境的软件。这些软件结合起来可以提供应用软件运行的平台。如操作系统、基于其上的数据库管理系统及开发工具、系统及网络管理软件、中文处理平台软件、图形图像管理软件、人机接口交互软件等。

②中间软件。主要包括中间软件和计算机安全软件产品。

③应用软件。应用软件指直接完成某种具体应用而无需用户重新编程的软件。应用软件根据实用范围的不同分为通用应用软件和行业应用软件。通用应用软件包括办公及文字处理软件、通用财务软件、教育软件、游戏娱乐软件等;行业应用软件涉及金融、邮电、政府、教育科

研、能源和交通等行业的特殊应用软件。

(3)按照软件规模分类

按开发软件所需的人力、时间以及完成的源程序行数,可将软件分为见表 1.1。

表 1.1　按软件规模进行的软件分类

类　别	开发人数/人	研发周期	程序规模(源程序行数)
微型	1	1 ~ 4 周	0.5 kB
小型	1	1 ~ 6 个月	1 ~ 2 kB
中型	2 ~ 5	1 ~ 2 年	5 ~ 50 kB
大型	5 ~ 20	2 ~ 3 年	50 ~ 100 kB
较大型	100 ~ 1 000	4 ~ 5 年	1 M(1 024 kB)
巨大型	2 000 ~ 5 000	5 ~ 10 年	1 ~ 10 M

(4)按软件工作方式进行分类

按软件工作方式,可将软件分为实时处理软件、分时软件、交互式软件、批处理软件。

①实时处理软件。实时处理软件指监视、监控、分析现实世界发生的事件,当事件或数据产生时,立即予以处理,并及时反馈信号、控制需要监控的过程的软件。主要包括数据采集、分析、输出 3 部分,其处理时间是被严格限定的,如果在任何时间超出这一限制,都将造成事故。

②分时软件。该类软件允许多个联机用户同时使用计算机,使系统把处理时间轮流分配给各联机用户,使各用户都感到只有自己在使用计算机。

③交互式软件。能实现人机通信的软件。该类软件接收用户的信息,但在时间上没有严格限定,这种工作方式给予用户很大的灵活。

④批处理软件。该类软件将一组输入作业或一批数据以成批处理的方式一次运行,按顺序逐个处理。

(5)按软件市场和标准化程度进行分类

从软件市场的角度考虑软件产品、软件服务和按软件的标准化程度,可以将软件分为标准化软件、半定制软件和服务软件。

①标准化软件。可以封装发售、买来即可使用的软件,如操作系统和办公软件。这类软件标准化强、销量大,价格相对较低,必须形成市场规模才能获得可观的利润。而且这类软件容易在市场上形成某种标准,如微软的 Office 办公软件现在已成为事实上的办公标准,其他办公软件必须与 Office 保持足够的兼容才能让最终的用户接受。

②半定制软件。具有相当一部分共性,但仍需要一定的客户化开发工作才能满足客户需要的软件。ERP 系统是一种典型的半定制软件。

③服务软件。必须根据特定客户的需求量身定制的软件。最典型的例子是系统集成服务。不存在通用的解决方案,软件的可复用性不强,软件企业必须针对每一位客户要求来编写代码,甚至可能是重新开发所有代码。

(6)其他分类方式

按软件运行在计算机网络中的客户端还是服务端,可以分为客户端软件和服务端软件。

1.2 软件危机

软件危机是指在计算机软件开发和维护时所遇到的一系列问题。软件危机主要包含两方面的问题:一是如何开发软件以满足社会对软件日益增长的需求,二是如何维护数量不断增长的已有软件。

(1)软件危机产生的原因

软件危机产生的原因与软件的特点有关,也与软件开发的方式、方法、技术和软件开发人员本身有关。

①软件不同于硬件,它是计算机系统中的逻辑部件而不是物理部件。在软件运行之前,软件开发过程的进展情况较难衡量,软件开发的质量也较难评价。因此,管理和控制软件开发过程相当困难。此外,由于软件缺乏"可见试"阶段,很可能是遇到在开发时期已引入,但在测试阶段没能检测出来的错误。因此,软件维护通常意味着改正或修改原来的设计,这就在客观上使得软件较难维护。

②软件一般要使用 5 ~ 10 年,在这段时间里,很可能出现在软件开发时没有预料到的问题。不仅必须改正使用过程中出现的每一个潜在的错误,而且当环境变化时(例如硬件或系统软件更新换代)还必须相应地修改软件以适应新的环境,特别是必须经常改进或扩充原来的软件以满足用户不断变化的需要。

③软件规模越来越大,结构越来越复杂,给软件的开发和维护带来客观的困难。

④软件开发技术落后,生产方式和开发工具落后,生产率提高缓慢。

⑤软件开发人员忽视软件需求分析的重要性,轻视软件维护,也是造成软件危机的原因。

(2)软件危机主要表现形式

①软件发展速度跟不上硬件的发展和用户的需求。软件应用日趋广泛,软件产品"供不应求"。随着微电子技术的进步和生产自动化程度的提高,硬件成本逐年下降,然而软件开发需要大量的人力资源,软件成本随着通货膨胀以及软件规模和数量的不断扩大而逐年上升。美国在 1995 年的调查表明,软件成本大约已占计算机系统总成本的 90%。

②对软件开发成本和进度的估计常常不准确,造成用户不满意。由于软件应用范围越来越广,很多应用领域往往是软件开发者不熟悉的,加之开发人员与用户之间信息交流不够,导致软件产品不符合要求或不能如期完成。因而,软件开发成本和进度都与原先的估计相差太大,引起用户不满。

③软件产品质量差,可靠性不能保证。软件质量保证技术没有应用到软件开发的全过程,导致软件产品质量问题频频发生。

④软件产品可维护性差。软件设计时不注意程序的可读性,不重视可维护性,造成程序中存在的错误很难改正。当软件需求发生变化时,维护却显得相当困难。

⑤软件没有合适的文档资料。软件开发时文档资料不全或文档与软件不一致,会引起用户不满,同时也会给软件维护带来很大的困难。

(3)解决软件危机的途径

软件危机是客观存在的,它既是软件本身的特殊性质造成的结果,也是人们对它认识不

足，还缺少有效的理论和技术驾驭或控制它导致的必然结果。多年来人们一直在努力研究寻找解决软件危机的途径。从目前的发展看，解决软件危机的途径主要有以下几种：

①充分吸取和借鉴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，特别是要吸取几十年来从事计算机硬件研究和开发的经验教训。

②推广使用在长期实践中总结出来的开发软件的成功的技术和方法，探索和研究更好的软件设计、表达技术及管理方法，消除计算机系统早期发展阶段形成的一些错误概念和做法；采用面向对象的软件设计方法，使解决问题的方法空间同客观世界的问题空间完全一致，降低或化解软件设计的复杂性。

③使用好的软件开发工具及软件工程环境。正如机械工具可以“放大”人类的体力一样，软件开发工具及软件工程环境可以“放大”人类的智力。在软件开发的各个阶段都有许多烦琐重复的工作需要做，在适当的软件工具的辅助下，可更好地辅助软件的生产，从而提高软件生产率，保证软件的质量。

④软件开发不是某种个体劳动，因此应该有良好的组织、严密的管理，促使各类人员相互配合共同完成任务。

为了解决软件危机，既要有技术措施（好的方法和工具），也要有组织管理措施。软件工程正是从技术和管理两方面来研究如何更好地开发和维护计算机软件的。

1.3　软件工程

“软件工程”这个概念是在1968年北大西洋公约（NATO）召开的一次专门会议上正式提出的。在这次大会上，与会的专家们认真地评估了软件危机对软件设计领域的危害及软件开发方法存在的种种问题，一致提出要致力于软件工程技术的发展，建议采用工程化的规范方法来指导软件的开发、运行及维护工作。

1.3.1　软件工程的概念

概括地说，软件工程是指导计算机软件开发和维护的一门工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，经济地开发出高质量的软件并有效地维护它，这就是软件工程。

1.3.2　软件工程的基本原理

著名的软件工程专家B. W. Boehm总结了TRW公司多年开发软件的经验，于1983年在一篇论文中提出了软件工程的7条基本原理。他认为这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

这是吸取前人的教训而提出来的。经统计表明，不成功的软件项目中有一半左右是由于计划不周造成的。在软件开发与维护的漫长生命周期中，需要完成许多性质各异的工作。这条原理意味着，应该把软件生命周期分成若干阶段，并制订出相应的切实可行的计划，然后严格按照计划对软件的开发和维护进行管理。Boehm认为，在整个软件生命周期中应指定并严

格执行6类计划: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。不同层次的管理人员都必须严格按照计划各尽其职管理软件开发与维护工作,绝不能受客户或领导的影响而擅自更改预定计划。

(2)坚持进行阶段评审

大部分错误是在编码之前造成的,错误发现与改正得越晚,所需付出的代价越高,相差大约2到3个数量级。因此,软件的质量保证工作不能等到编码结束之后再进行,应坚持进行严格的阶段评审,以便尽早发现在软件开发过程中的错误。

(3)实行严格的产品控制

在软件开发过程中不要随意改变需求。因为改变某项需求往往需要付出较高的代价,但在实践中用户往往会提出需求变更,因此需要采取科学的产品控制技术。目前主要实行基准配置管理,基准配置是指经过阶段评审后的软件配置成分,如各个阶段产生的文档或程序代码。对涉及基准配置的修改,必须经过严格的评审,通过后才能实施修改。

(4)采纳现代程序设计技术

从20世纪六七十年代的结构化软件开发技术,到面向对象技术,从第一、第二代语言,到第四代语言,人们已经充分认识到:方法比力气更有效。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,开发过程难以评价和管理。为更好地进行管理,应根据软件开发的总目标及完成期限,规定开发组织的责任和产品标准,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

开发人员的素质和数量是影响软件质量和开发效率的重要因素,应该少而精。这一条基于两点原因:高素质开发人员的效率比低素质开发人员的效率要高几倍到几十倍,开发工作中犯的错误也要少得多;当开发小组为 N 人时,可能的通信信道为 $N(N-1)/2$,可见随着人数 N 的增大,通信开销将急剧增大。

(7)承认不断改进软件工程实践的必要性

遵从上述6条基本原理,就能够较好地实现软件的工程化生产。但是,上述6条原理只是对现有的经验的总结和归纳,并不能保证赶上技术不断前进发展的步伐。因此,Boehm提出应把承认不断改进软件工程实践的必要性作为软件工程的第7条原理。根据这条原理,不仅要积极采纳新的软件开发技术,还要注意不断总结经验,收集进度和消耗的数据等,进行出错类型和问题报告统计。

1.3.3 软件工程研究的内容

软件工程包括技术和管理两方面的内容,是技术与管理紧密结合所形成的工程学科。软件工程的研究内容如图1.1所示。

软件工程学科具有以下几个特点:

(1)研究内容的综合性

软件工程作为指导软件开发、运行、维护的工程性学科,它研究内容涉及面广,具有多学科性。所谓多学科,指软件工程不仅包含与软件有关的课题,还涉及其他许多学科,如管理科学、