



软件工程

马林艺 张喜英 钱春升 编著



增值回报
电子教案



21 世纪高职高专规划教材系列

软 件 工 程

马林艺 张喜英 钱春升 编著



机 械 工 业 出 版 社

本书从基本概念到理论、实践,对软件工程及其相关知识进行了详细的阐述,并列举了大量的实例。

全书共分10章,主要内容包括:软件工程的概述,需求工程,面向对象开发方法,软件设计方法,面向对象软件设计,软件的编码,软件测试,软件进化,软件项目管理,案例分析与课程设计。书中配有习题,网上可下载电子教案。

本书可作为高职高专计算机及相关专业的教材和参考书,也可作为毕业设计指导用书。

图书在版编目(CIP)数据

软件工程/马林艺等编著. —北京:机械工业出版社, 2006.1

(21世纪高职高专规划教材系列)

ISBN 7-111-17601-4

I. 软... II. 马... III. 软件工程—高等学校:技术学校—教材

IV. TP311.5

中国版本图书馆CIP数据核字(2005)第120896号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策 划:胡毓坚

责任编辑:王 颖

责任印制:洪汉军

北京原创阳光印业有限公司印刷

2006年1月第1版·第1次印刷

787mm×1092mm 1/16·15.25印张·384千字

0001-5000册

定价:22.00元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68326294

封面无防伪标均为盗版

出版说明

为了贯彻国务院发[2002]16号文件《国务院关于大力推进职业教育改革与发展的决定》的精神,进一步落实《中华人民共和国职业教育法》和《中华人民共和国劳动法》,实施科教兴国战略,大力推进高等职业教育改革与发展,我们组织力量,对实现高等职业教育培养目标和保证基本教学规格的文化基础课程、专业技术基础课程和重点建设专业主干课程的教材进行了规划和编写。

本套教材内容涵盖了高职高专院校计算机及相关专业的专业基础课、专业课以及选修课程,主要分为计算机文化基础、编程语言、硬件技术、网络信息、数据库应用及多媒体技术等几大类。为配合高职教育关于“培养 21 世纪与我国现代化建设要求相适应的一线科技实用型人才”的最新理念,我们特为本系列教材配备了实践指导丛书,以利于老师的教学和学生的学习。

本套教材将理论教学和实践教学紧密结合,图文并茂、内容实用、层次分明、讲解透彻,其中融入了作者长期的教学经验和丰富的实践经验。可作为各类高职高专院校的教材,也可作为各类培训班的教材。

机械工业出版社

前 言

随着我国软件行业的日益兴起,社会对软件人才的需求也呈现高速增长的趋势。然而,软件人才在数量上的缺乏以及能力水平的不足,已经成为制约我国软件行业发展的一个瓶颈。究其原因,主要在于人们片面重视编程方法的掌握,却忽视了工程项目的整体把握,使得软件行业陷入一种危机中,这就是人们担忧的“软件危机”。

软件工程是一项用于指导软件开发和维护的新技术,它综合了计算机科学、数学和管理科学等多个学科领域的研究成果,对解决“软件危机”起到了不可忽视的重要作用。

软件工程的主要内容包括:支持软件开发与维护的理论、方法、技术、标准,以及计算机辅助软件工程等。软件工程具有一些显著的特点,如技术方法的抽象化、形式化以及工程项目的复杂性、事物繁琐等。由于软件工程的上述特点,使人们在掌握和理解软件工程的相关知识方面有一定的难度。

本书作者一直从事软件工程的教学与研究,根据多年的经验编写了本书,旨在进一步普及软件工程学,使读者对软件工程学有一个全面的认识,进而提高读者的软件开发能力。

为了编写本书,作者做了大量的调查研究。在人才需求方面,了解到软件行业最需要这样的软件应用人才:

- 1) 能提取和分析出好的需求。
- 2) 能制定良好的开发原则。
- 3) 能正确地使用 UML。
- 4) 能将需求与设计转化为有效的代码。
- 5) 能测试并完善这些代码。

在软件工程类书籍方面,目前国内的大多数书籍把软件工程截然分成“面向过程”和“面向对象”两大部分。事实上,当今一个软件项目的开发,是把面向过程、面向对象甚至更新的技术作为一个整体来考虑的。

根据上述调查结果,作者制定了本书的编写原则:

- 1) 具备开拓创新的思路,理论和实践相结合。
- 2) 每介绍一个新的知识点,都要举一个生动的示例,并进行相应的练习和实习的描述,帮助读者了解和掌握相关的知识。
- 3) 以循序渐进的方式,引导读者在理论上和实际上接近或达到软件行业的需求目标。
- 4) 借鉴国外相关的技术方法,从需求、设计、实现、项目管理等方面展开综合论述,力图在技术、方法上与国际接轨。
- 5) 考虑到软件工程的实践指导类书籍较少,最后一章通过分析实例并结合课程设计,提供一套实践指导内容。附录列出了各章相关的文档标准和规范。

全书共分 10 章,第 1、2、3 章由张喜英编写,第 4、5、6、7 章由钱春升编写,第 8、9、10 章及附录由马林艺编写。全书由马林艺统稿。由于作者水平有限,书中存在的不足和错误,请读者批评指正。

本书涉及知识范围广泛、论述简洁明了、实例丰富,可作为高职高专院校的教材或参考书,

也可供选择软件设计毕业课题的毕业生、有一定经验的软件工作人员和致力于软件项目开发的广大计算机用户阅读参考。

为了配合本书的教学,机械工业出版社为读者提供了电子教案,读者可在 www.cmpbook.com 上下载。

作 者

目 录

出版说明

前言

第 1 章 概述	1
1.1 软件工程介绍	1
1.1.1 软件与软件危机	1
1.1.2 软件工程	4
1.1.3 基于计算机的系统工程	6
1.1.4 传统软件工程与面向对象软件工程	10
1.2 软件工程过程及生命周期	11
1.2.1 软件工程过程	11
1.2.2 软件生命周期	12
1.3 软件开发模型	14
1.3.1 传统开发模型	14
1.3.2 演化模型	17
1.3.3 面向对象模型	19
1.3.4 形式化模型	20
1.4 软件项目管理	21
1.4.1 软件项目计划	22
1.4.2 项目管理	23
1.4.3 软件质量管理	24
1.5 习题	25
第 2 章 需求工程	26
2.1 软件需求	26
2.1.1 需求分析	26
2.1.2 获取需求的方法	31
2.2 需求工程过程	33
2.2.1 需求工程步骤	33
2.2.2 需求工程常用分析模型工具	35
2.3 系统模型	40
2.3.1 两种分析模型	40
2.3.2 结构化建模技术	41
2.3.3 结构化分析方法	49
2.4 需求与软件原型	53
2.4.1 软件过程中的原型开发	53
2.4.2 快速建立软件原型	55

2.5 需求说明书	57
2.5.1 软件需求规格说明	57
2.5.2 需求评审	59
2.6 习题	59
第3章 面向对象开发方法	61
3.1 面向对象方法概述	61
3.1.1 面向对象方法学要点	61
3.1.2 面向对象方法的优势	62
3.2 面向对象的基本概念	64
3.2.1 对象	64
3.2.2 其他概念	65
3.3 面向对象建模	68
3.3.1 面向对象的需求分析	68
3.3.2 统一建模语言——UML	70
3.3.3 面向对象建模技术	72
3.4 面向对象分析过程	77
3.4.1 面向对象分析的基本概念	77
3.4.2 定义用例	79
3.4.3 领域分析	80
3.4.4 类/对象建模	80
3.4.5 建立对象关系模型	83
3.4.6 建立对象行为模型	84
3.5 习题	85
第4章 软件设计	86
4.1 软件设计的基本概念	86
4.2 软件设计原则	88
4.2.1 模块独立性的含义	88
4.2.2 模块的耦合性	89
4.2.3 模块的聚合性	90
4.2.4 模块的规模	91
4.3 软件的其他原则	92
4.3.1 信息隐藏和局部化的原则	92
4.3.2 抽象的原则	92
4.3.3 控制层次适中的原则	92
4.4 概要设计	93
4.4.1 概要设计的过程	93
4.4.2 概要设计使用的工具	95
4.4.3 软件设计方法	97
4.5 详细设计	100

4.5.1 详细设计的任务	100
4.5.2 详细设计使用的工具	100
4.5.3 Jackson 程序设计方法	104
4.6 设计规格说明书与设计评审	104
4.6.1 软件概要设计说明书大纲	104
4.6.2 软件详细设计说明书大纲	105
4.6.3 设计评审	105
4.7 习题	105
第5章 面向对象软件设计	107
5.1 面向对象设计概述	107
5.1.1 面向对象设计的任务	107
5.1.2 面向对象的设计模型	107
5.2 系统设计	111
5.2.1 系统设计过程	111
5.2.2 子系统设计	112
5.2.3 用户界面设计	115
5.2.4 任务管理设计	116
5.2.5 数据管理设计	117
5.3 对象设计	118
5.3.1 对象描述	118
5.3.2 算法设计	119
5.3.3 程序构件与接口	119
5.4 领域对象设计	120
5.4.1 领域对象的设计内容	120
5.4.2 领域对象的设计模板	121
5.5 习题	121
第6章 软件的编码	123
6.1 结构化程序设计	123
6.1.1 结构化程序设计的原则	123
6.1.2 程序设计自顶向下、逐步求精	124
6.1.3 数据结构的合理化	125
6.2 面向对象编程	125
6.3 程序设计的风格	126
6.3.1 源程序文档化	126
6.3.2 数据说明	127
6.3.3 语句结构	128
6.3.4 输入和输出	129
6.4 程序设计语言	130
6.4.1 程序设计语言的分类	130

6.4.2 设计语言的技术特征及选择方法	130
6.5 编程的标准和过程	131
6.5.1 制定编程标准	131
6.5.2 编程阶段的三大任务	132
6.6 习题	134
第7章 软件测试	136
7.1 软件测试的概念与原则	136
7.1.1 测试的概念	136
7.1.2 测试过程	138
7.1.3 测试的原则	138
7.2 软件测试技术	139
7.2.1 软件测试的目标	139
7.2.2 测试方法	140
7.3 软件测试策略	142
7.3.1 单元测试	142
7.3.2 组装测试	144
7.3.3 确认测试	145
7.3.4 系统测试	146
7.4 软件纠错技术	146
7.4.1 纠错的策略	146
7.4.2 纠错技术	147
7.5 面向对象的软件测试	147
7.5.1 面向对象测试策略	148
7.5.2 面向对象软件测试用例设计	149
7.5.3 类间测试用例设计	150
7.6 测试计划和测试分析报告	150
7.6.1 测试计划	150
7.6.2 测试分析报告	151
7.7 习题	152
第8章 软件进化	154
8.1 遗留系统	154
8.1.1 遗留系统的结构	154
8.1.2 遗留系统设计	155
8.1.3 遗留系统评估	156
8.2 软件维护	157
8.2.1 软件维护的概念	158
8.2.2 维护任务的实施	159
8.2.3 软件维护性与软件质量	160
8.2.4 维护的代价及其主要因素	161

8.3 软件再工程	162
8.3.1 重构	162
8.3.2 逆向工程和前向工程	163
8.3.3 软件重用技术	164
8.4 配置管理	165
8.4.1 配置管理概念	165
8.4.2 软件配置标识	166
8.4.3 变更管理	166
8.4.4 配置审核	168
8.4.5 配置状态报告	169
8.4.6 版本和发布管理	169
8.5 习题	170
第9章 软件项目管理	171
9.1 概述	171
9.1.1 项目和项目管理定义	171
9.1.2 软件工程项目管理的特点和任务	172
9.1.3 项目的计划与组织	173
9.2 人员管理	174
9.2.1 项目的人员组成	174
9.2.2 项目的领导	175
9.2.3 团队协作	176
9.2.4 人员能力成熟度模型	177
9.3 软件开发成本估算	178
9.3.1 生产率	179
9.3.2 估算技术	179
9.3.3 算法成本建模	180
9.3.4 项目的进度安排和人员配备	181
9.4 软件质量管理	182
9.4.1 质量保证和质量标准	182
9.4.2 质量规划	184
9.4.3 质量控制	185
9.5 软件过程改进	185
9.5.1 软件过程 and 产品质量	186
9.5.2 软件过程改进的步骤	187
9.5.3 软件过程度量	188
9.5.4 软件过程改进模型——能力成熟度模型(CMM)	188
9.6 习题	190
第10章 案例分析与课程设计	191
10.1 高校工资管理系统设计与实现	191

10.1.1	系统分析	191
10.1.2	系统设计	200
10.1.3	系统实现	202
10.1.4	系统测试与维护	202
10.2	其他信息系统设计案例	203
10.2.1	计算机自动出卷系统	203
10.2.2	图书馆藏书借阅管理系统	205
10.2.3	商品进销存系统	206
10.3	课程设计与论文规范	207
10.3.1	课程设计	207
10.3.2	毕业设计(论文)撰写规范	209
附录		212
附录 A	不同规模软件设计所需文档一览表	212
附录 B	计算机软件产品开发文件编制指南(GB/T8567—1988)	213
参考文献		231

第1章 概述

本章要点

- 本章介绍软件的基本概念、软件的特点及软件的发展与软件危机,讲述了软件工程的出现、软件工程学的有关知识和一些基本概念。
- 介绍软件工程过程的一系列相关活动,讲解了软件生存周期要经历软件定义、软件开发、软件维护的全过程。
- 介绍主要的软件过程模型,包括传统模型、演化模型、面向对象模型、形式化模型等。
- 介绍贯穿于整个软件生存周期——项目进度计划、项目管理及软件质量管理等有关管理工作的内容。

1.1 软件工程介绍

1.1.1 软件与软件危机

1. 软件的概念、特点

很多人把软件理解为计算机程序,这种理解是比较狭隘的,计算机程序并不就是计算机软件,那么什么是计算机软件呢?软件属于计算机系统逻辑部件,是计算机系统的不可缺少的组成部分。计算机的运算器、控制器、存储器以及各种外部设备都是看得见的东西,我们称之为硬件;从计算机系统中除去实体的各部件所剩下的所有部分均属于软件。软件比任何其他人类制造的结构更复杂,甚至硬件的复杂性和软件比起来也是微不足道的。

软件的发展,大体上经历了程序、程序系统和软件等3个阶段。早期的程序规模较小,随着系统程序的增加,人们把程序区分为系统程序和应用程序,但是这两个阶段在开发过程中都很少或无法考虑到它们的维护,随着程序规模不断增大,需求量不断增加,个人编程已经无法适应,需要有许多人分工协作,或者委托他人才能完成编程的任务。由此对程序的描述、分工说明、各种技术约定以及程序管理与维护的规则等,就成为不可缺少的要求。软件工程兴起后,人们把软件视为产品,强调软件的“可维护性”,同时确定了各个开发阶段必需完成的“文档”。一个软件系统通常包含大量独立的程序、用于设置这些程序的配置文件、描述系统结构的系统文档和如何使用该系统的用户文档,以及告知用户下载最新产品信息的Web站点。

软件是一种抽象的、逻辑性的产品,软件是程序和所有使程序正确运行所需要的相关文档和配置信息。一个计算机系统要正常的工作,充分发挥其硬件的各种功能,必须配备完善的软件。

软件应具有这样几个特点:

(1) 软件是计算机系统的一个部分

若用户直接使用硬件构成的、未配备任何软件的计算机,只能利用该计算机提供的机器指令进行操作,那是非常困难的,而且也限制了计算机在社会各个领域的广泛应用;软件是使用计算机时不可缺少的部分,软件需要和机器保持一致,现实中往往不会是机器适应软件,而是

软件的使用和硬件有直接的关系并受硬件的限制,因为人们常认为软件是最容易调整的部分,也就造成了有的软件对硬件的依赖性非常大,可能只能在某种型号的计算机上使用,甚至依赖于某种操作系统。基于以上两个方面,在软件开发时必须考虑可移植性,这也是衡量软件质量的一个重要指标。

(2) 软件是一种抽象的、逻辑性的东西

软件不像硬件能看得见、有磨损,它的使用寿命完全决定于它的应用需要,只要有用户,它并不存在损耗,可以无限制地被使用;实际软件也存在使用寿命,在它的存在过程中,可能需要不断地进行修改、维护,以适应变化的硬件环境、软件环境和用户的新的需求,也可能需要不断地改正正在开发过程中忽视的,或者错误的一些东西。当软件的维护成本大于开发新软件的成本时,将是软件寿命终止的时候。

(3) 软件可以作为商品出售

作为商品来说软件价值可以从两个方面来衡量:一是开发软件所消耗的劳动量的价值,另一方面是使用软件所能带来的价值。无论从哪个角度来衡量,软件只有走到商品化和产业化的道路上,才有可能克服软件开发和应用所面临的复杂性难以度量、成本居高不下、维护困难等缺陷。

(4) 软件的应用与发展与社会的进步密切相关

软件的应用,特别是在经济生活领域的应用,与人们的观念和知识要求紧密联系,尤其体现在管理信息系统软件的开发和应用上,如果运行机制和管理方式及人的心理认同程度不够,肯定会影响到软件的开发和项目的成功。

2. 软件的分类

到目前为止,人们开发了大量的软件,积累了丰富的软件资源,用于各个方面的、功能性能各异的软件成千上万,要想给软件做出科学的分类是很难的,一般从功能上来看,计算机软件总体上划分两类:一类是系统软件,一类是应用软件。

- 系统软件。当计算机在执行各类信息处理任务时,那些管理与支持计算机系统资源及操作的程序,称为系统软件。
- 应用软件。那些综合用户信息处理需求的,直接处理特定应用的程序称为应用软件。

图 1-1 综合了各种软件的主要功能类型。系统软件主要依赖于计算机系统的制造商及计算机系统的型号,它是为管理、控制与监督、维护计算机及外设的工作,以及提供计算机与用户界面等的软件。对于终端用户来讲,应用软件是大量的,它主要用于解决一些实际应用问题,在系统软件的基础上应用软件配置得怎样,直接影响到计算机系统资源的效能。

3. 软件的发展和软件危机

20 世纪 40 年代中期出现了世界上第一台计算机,就产生了程序设计的概念。其后经过几十年的发展,计算机软件经历了 3 个发展阶段:

- 1) 程序设计阶段,大约在 20 世纪 50、60 年代。
- 2) 程序系统阶段,大约在 20 世纪 60、70 年代。
- 3) 软件工程阶段,大约在 20 世纪 70 年代以后。

早期的软件主要指程序,程序的开发采用个体工作方式,开发工作主要依赖于开发人员的个人技能和程序设计技巧。当时的软件规模小,开发过程不规范,缺少与程序有关的文档,软件开发的实际成本和进度往往与预计的相差甚远,软件的质量得不到保证,开发出来的软件常

常不能使用户满意。

随着计算机应用的需求不断增长,软件的规模也越来越大,长达数万行、数十万行乃至百万行以上的软件,已不鲜见,个人编程已经适应不了需要,需要由多人分工合作来完成编程任务。软件规模的增长,带来了它的复杂度的增加,如果说编写一个数十到数百行的程序能很轻易地完成,则开发一个数万以至数百万行的软件,其复杂度将大大上升,即使富有经验的人员也可能顾此失彼。软件的复杂性使软件产品很难理解,即使是开发队伍中的人员,可能对自己所负责的部分非常熟悉,但不可能准确地理解软件的所有部分;软件的复杂性不仅影响软件产品本身,而且也会影响软件过程管理,除非管理人员能得到他管理工程的精确信息,这一点实际上是无法做到的,结果可能使软件的开发费用经常超支,完成时间也经常脱期。更为严重的是随着软件规模的增长,可靠性在下降,质量保证也越来越困难,导致软件开发的生产率远远跟不上计算机应用的迅速增长。

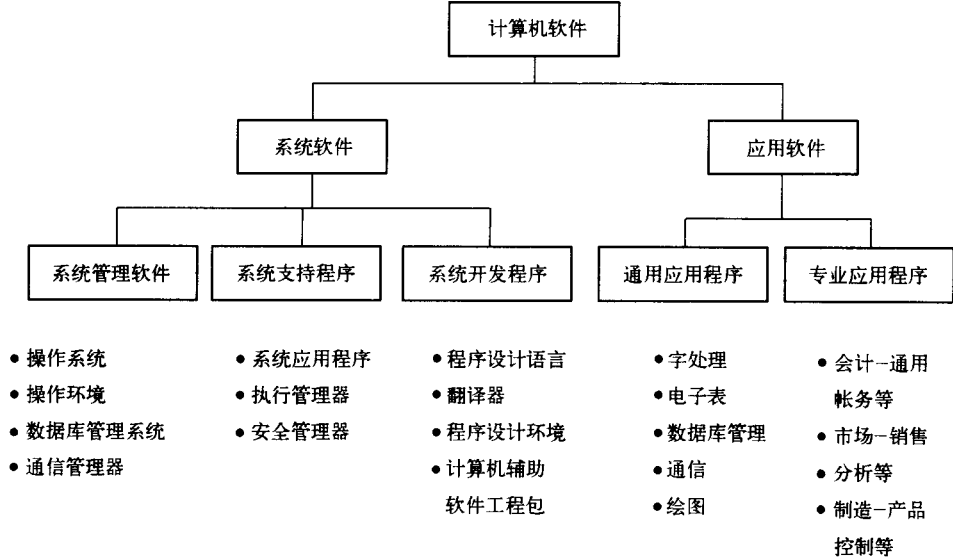


图 1-1 计算机软件分类

此外由于软件开发时缺少好的方法指导和工具辅助,同时又缺少有关的文档,使得大量已有的软件难以维护。一个大型软件,即使在开发时经过严格的测试与纠错,也不能保证运行中不再出现错误。不仅如此,在软件的生存周期中,还可能会因为功能的变化和环境的改变而需要修改软件,不言而喻规模大的软件,维护的成本必然上升。

上述这些问题严重地阻碍了软件的发展。20 世纪 60 年代中期,人们把上述软件开发和维护中的各种问题称为“软件危机”。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。具体表现形式如下:

- 1) 对软件开发成本和进度的估算很不准确。由于在开发过程中,用户需求在变动,开发方法落后,缺乏有效的开发过程管理,导致开发期延长,不能按时完成任务,使开发成本和进度与原先的估计相差悬殊,而为了赶进度、节约成本又损害了软件的质量。
- 2) 用户对完成的软件很不满意。软件开发人员对用户的需求缺乏深入的理解,有时甚至对要解决的问题还没有足够的认识,就急于编程,没有规范化的开发过程指导,加上开发人员

与用户的交流不够,必然导致开发的产品不能满足用户的需要。

3) 软件产品的质量很不可靠。在软件开发过程中,没有有效的管理手段对开发过程约束,缺乏严格的质量控制程序,而且也没有对软件质量确切的定量概念和评价标准,导致开发的软件质量难以保证,可靠性差。

4) 没有完整的文档。由于开发人员只注重算法和程序的实现,软件产品缺乏必要的文档资料或者文档资料不合格,使日后的使用和维护增加了许多困难和问题。

5) 软件成本比重上升。随着软件应用面的扩大,软件的花费越来越大,在整个计算机系统中所占的比重也越来越高,软件、硬件成本变化趋势如图 1-2 所示。

6) 软件开发生产效率低下,跟不上计算机应用迅速深入的趋势。硬件成本逐年下降,软件应用日趋广泛,但生产方式长期没有突破手工业的方式,软件开发的自动化程度低,技术进步落后与需求的增长,造成“供不应求”的局面,如图 1-3 所示。

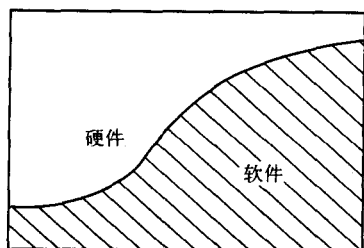


图 1-2 硬件/软件成本变化趋势

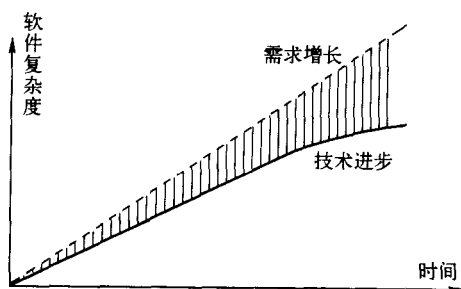


图 1-3 软件技术进步落后于需求的增长

解决软件危机的途径必须从软件开发的工程化方法入手,用现代工程的概念、原理、技术和方法去指导软件的开发、管理和维护,这就是软件工程的思想和方法。

1.1.2 软件工程

软件工程尝试解决交付期的推迟、预算超出或软件充满残余故障等问题。1968 年,北大西洋公约组织(NATO)的计算机科学家在德国召开国际会议,讨论“软件危机”问题,在这次会议上正式提出并使用了“软件工程”这个名词,希望用工程化的原则和方法来克服软件危机,一门新兴的工程学科诞生了。

1983 年,IEEE 给软件工程下的定义是:软件工程是开发、运行、维护和修复软件的系统方法。这个定义强调软件工程是系统方法而不是一种个人“艺术”。

FAIRLY 给出的定义是:“软件工程学是为了在成本限额以内按时完成和修改软件产品所需要的系统生产和维护技术及管理学科。”它明确指出软件工程的目的是在成本限额以内按时完成和修改软件产品的工作,同时指出软件工程应该包括技术和管理两个方面的内容。

FRITZ BAUER 认为:“软件工程是为了经济地获得可靠的且能在实际机器上有效运行的软件,而建立和使用完善的工程化原则的一门工程学科。”这个定义指出了软件工程的目的是经济地开发出高质量的软件,强调软件工程是一门工程学科,它必须建立和使用完善的工程化原则。

1993 年,IEEE 进一步给出了软件工程的更全面的定义:①软件工程是把系统化、规范的可度量的途径应用于软件开发、运行和维护的过程,即把工程化应用于软件生产中;②研究上

面第一点提出的途径。这个定义强调了工程化方法和如何实现这些方法在软件工程中的重要地位。

我国在计算机科学技术百科全书给软件工程的定义是:软件工程是“应用计算机科学、数学及管理科学等原理,开发软件工程。”并指出它借鉴传统工程的原则、方法,以提高质量,降低成本为目的。

1. 软件工程的基本原理及过程

根据以上软件工程的定义,我们可以综合出如下定义:

采用工程的概念、原理和方法来开发和维护软件,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,这就是软件工程。软件工程是一门工程学科,涉及软件生产的各个方面,从最初的系统描述一直到使用后的系统维护,都属于其学科范畴。研究的目的在于生产无故障的、及时交付使用的、在预算内的软件产品。

著名的软件工程专家 B.W.Boehm 在总结其他学者的意见并结合多年的开发工作经验,在 1983 年的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。完善的计划是软件项目成功的前提,在漫长的软件开发过程中,应该把软件生命周期划分成若干个阶段,并相应地制定出切实可行的计划,然后严格按照计划对软件的开发与维护工作进行管理。

2) 坚持进行阶段评审。

3) 实行严格的产品控制(推迟实现的原则)。

4) 使用现代程序设计技术。

5) 结果应能清楚地审查。

6) 开发小组的人员应少而精。

7) 承认不断改进软件工程实践的必要性。

2. 软件工程的目标与原则

软件工程就是从管理和技术两个方面研究如何更好地开发和维护计算机软件的一门学科,如何组织和实施软件工程项目。技术和管理是达到目标的手段,项目的成功是最终的追求。衡量成功有以下几个指标:

1) 能够满足基本需要。

2) 开发成本要小。

3) 较低的维护费用。

4) 及时完工并交付使用。

5) 可移植性好。

更具体的产品性能指标是,开发的软件产品应该具有可修改性、有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可互操作性。

软件工程的目标为软件开发提出了明确的要求。为了达到这些要求,在软件开发中必须遵循下列软件工程的原则:抽象、信息隐蔽、模块化、局部化、一致性、完整性和可验证性。

3. 软件工程包括 3 个要素——方法、工具和过程

软件工程方法为软件开发提供了“如何做”的技术。它包括项目计划与估算、软件系统需求分析、数据结构设计、系统总体结构设计、算法过程设计、编码、测试以及维护等。