

软 件 工 程

周 枫 刘晓燕 李秀敏 李 丹 编著

重 庆 大 学 出 版 社

内 容 简 介

软件工程是介绍软件开发方法,提高学生软件开发能力的一门重要的专业课程。本教程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。全书共分12章,内容包括软件工程概论,软件计划,软件需求分析,软件结构设计,软件详细设计,编码,软件测试,面向对象的软件开发技术,软件维护,软件质量保证,软件工程环境,软件工程新进展。本教材内容新颖、全面,对软件开发具有指导性作用。

本教程主要作为大专院校计算机本科专业的软件工程课程教材或研究生考试的复习辅助教材,也可以作为计算机专业技术人员学习考试的参考教材。

图书在版编目(CIP)数据

软件工程/周枫等编著. —重庆:重庆大学出版社,2001.7

计算机科学与技术本科系列教材

ISBN 7-5624-2300-8

I. 软... II. 周... 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(2001)第25012号

软 件 工 程

周 枫 刘晓燕 李秀敏 李 丹 编 著

责任编辑 曾令维 穆安民

*

重庆大学出版社出版发行

新华书店经销

重庆大学建大印刷厂印刷

*

开本:787×1092 1/16 印张:14.75 字数:368千

2001年7月第1版 2001年7月第1次印刷

印数:1—6000

ISBN 7-5624-2300-8/TP·281 定价:22.00元

前言

随着信息技术的发展,计算机已被广泛应用于社会的各个方面,成为推动社会发展的技术动力。而在计算机应用中,软件的作用十分突出,软件已经发展成为信息技术的核心,主导着信息产品的开发和信息技术市场的进一步开拓。软件产业已经成为社会信息化进程中的一个战略性产业。1998 年全球软件产值超过千亿美元,2000 年达到 2 千亿美元。

在软件技术的应用中,软件的开发技术,尤其是应用型软件产品的开发技术成了重中之重。不断开发适应用户需求、市场需要的新型软件产品,参与市场竞争,获取最大利润是全球信息企业追求的目标。而软件产品的竞争,说到底,就是人才的竞争、技术的竞争。因此,如何树立良好的软件开发理念,建立正确的、符合工程规范的、系统化的软件工程思想,灵活运用各种软件分析与设计技术,开发高质量的软件产品,是一个软件开发人员应当具备的基本技术素质。

软件工程是计算机科学与技术专业开设的,专门介绍软件开发技术思想、方法和应用的专业课程。该门课程以支持软件生存周期的软件开发模型为主导,系统、全面地介绍软件工程学科的概念、原理、方法、软件工具及支持环境。在内容上,它综合了计算机科学与技术、工程学、数学、管理学等几大学科的内容;在方法上,它涉及软件的计划、分析、设计、编程、测试和维护整个生存周期。内容纵横交错,方法前后关联。不仅具体支持了软件开发管理工作,还极大地丰富和发展了软件技术理论和计算机应用技术。

本书内容共分 12 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,这些概念构成了软件工程原理与方法的基础;第 2 章 软件计划,主要介绍软件项目可行

性分析,以及如何制订科学的软件开发计划;第3章 需求分析,主要介绍如何根据需求信息的分析建立软件系统逻辑模型的过程;第4章 软件结构设计,主要介绍软件设计的概念、原理、方法及具体运用的过程;第5章 详细设计,主要介绍了结构化的算法设计的思想及描述工具;第6章 编码,第7章 软件测试,这两章主要从技术实现的角度,介绍了语言的特性、选择的原则及测试一个软件系统所采用的各种测试方法;第8章 面向对象的软件开发技术,主要介绍面向对象的概念、对象分析、对象设计及实现的方法,并以 ATM 系统为例具体讲述其应用;第9章 软件维护,介绍了软件维护的有关概念和方法;第10,11,12章 分别介绍了软件质量保证、软件工程环境及软件工程新进展的内容。

本书力求做到基础内容与新技术并重、理论与实际应用相融合,并注意与软件技术中其他课程的衔接。通过该门课程的学习,旨在训练学生的软件分析思维能力,使学生进一步掌握软件系统的各种开发方法,提高软件设计水平,为以后的软件开发工作打好基础。

本书第1,8,10,11,12章由周枫编写,第2,3章由刘晓燕执笔,第4,5章及第8章的例题部分由李秀敏执笔,第6,7,9章由李丹执笔。全书由周枫统稿。有关的老师和同事参与了大纲的讨论,在此一并表示感谢。

限于编者的学识和经验,书中难免有差错和遗漏,恳请读者不吝指正。

编 者

2001年3月

目 录

第1章 软件工程概论	1
1.1 计算机系统的发展	1
1.2 计算机系统工程	2
1.3 软件的特征	4
1.4 软件危机	6
1.5 软件工程的概念	8
1.6 软件工程中的五种软件开发模式	12
1.7 软件工具、环境及应用平台	16
习题	18
第2章 软件计划	19
2.1 软件可行性分析	19
2.2 软件计划	21
2.3 软件的成本估算	24
2.4 安排开发进度	29
2.5 软件计划说明书	31
习题	31
第3章 软件需求分析	33
3.1 需求分析的任务和步骤	34
3.2 结构化分析方法	37
3.3 原型化分析方法	47
3.4 需求分析工具及进展	51
3.5 需求分析的文档及复审	52
习题	54
第4章 软件结构设计	56
4.1 结构设计的目标及任务	57
4.2 结构化设计的概念和原理	60

4.3	面向数据流的设计方法	66
4.4	面向数据结构的设计方法	78
4.5	设计规格说明书及设计复审	90
	习题	92
第5章	详细设计	94
5.1	详细设计的目标及任务	94
5.2	算法的结构化设计思想	95
5.3	算法的描述工具	97
5.4	算法的复杂性度量	104
	习题	108
第6章	编码设计	110
6.1	编码设计的目标与任务	110
6.2	编码的工具语言	111
6.3	编码的风格	117
	习题	120
第7章	软件测试	121
7.1	软件测试的目标及原则	121
7.2	软件测试的概念	122
7.3	测试实例的构造	126
7.4	测试方案的建立及实施	135
7.5	软件排错(Debug)	141
	习题	145
第8章	面向对象的软件开发技术	147
8.1	面向对象的概念	148
8.2	面向对象的开发模型	152
8.3	面向对象分析	154
8.4	面向对象设计	172
8.5	面向对象实现	180
	习题	184
第9章	软件维护	186
9.1	软件维护的目标与任务	186
9.2	软件维护的类型	187
9.3	软件的可维护性	189
9.4	软件维护方法	190
9.5	软件维护的管理	192
	习题	196

* 第 10 章 软件质量保证	197
10.1 软件质量的概念	197
10.2 软件质量保证方法	201
10.3 软件工程标准介绍	205
习题	207
* 第 11 章 软件工程环境	208
11.1 软件工程环境的概念	208
11.2 软件工程环境的结构	210
11.3 软件工程环境中的应用技术	212
11.4 软件工程环境实例介绍	215
习题	217
* 第 12 章 软件工程新进展	218
12.1 软件集成化技术	218
12.2 软件形式化技术	221
12.3 软件智能化技术	223
习题	225
参考文献	226

第 1 章

软件工程概论

计算机技术是 20 世纪的一项伟大发明。究其技术的发展过程可以归结为一门广义的学科——计算机系统工程。其中又可以分为硬件系统工程、软件系统工程。它们共同的目标是,为了开发各种以计算机为基础的应用系统时引入某种规律性。

计算机硬件工程是从微电子设计技术中逐步发展起来的。在近 50 年的发展过程中已经达到较成熟的水平。硬件的设计技术已经确立,其设计的自动化水平及效率较高。目前计算机技术发展的重点是不断改进制造工艺,以更好的性能价格比为计算机应用提供硬件支持。硬件系统的可靠性已经是一种可以期待的现实。硬件产品已经能从功能上、性能上及数量、质量上不断满足计算机应用的需求。

计算机软件在今天已经取代硬件成为计算机系统中设计最困难、最不容易获得成功(成功是指:按时完成和不超过预算成本并能满足用户需求),而管理上最冒风险的部分。随着技术的发展,计算机应用需求的日益扩大,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。软件系统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。

1.1 计算机系统的发展

计算机系统的发展以 1946 年研制成功的 ENIVAC 电子管计算机为标志,经历了第一代的电子管数字计算机、第二代的晶体管数字计算机、第三代的集成电路计算机、第四代大规模集成电路计算机,以及今天广为人知的新一代计算机(又称为第五代计算机)等几个阶段。几十年的发展,硬件的设计技术、制造工艺日趋成熟。更优的硬件性能、更小的体积、更低的硬件成本推动了软件技术的发展。反过来又从更深的层次上推动硬件自身的发展。对于硬件技术的发展,我们不过多叙述,而主要介绍软件技术的发展进程。

软件技术的发展最初是伴随着硬件技术而发展的。其发展大致可以分为四个阶段:

① 程序时代(20 世纪 60 年代以前)。早期的计算机系统中,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,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(Art)阶段。

这个阶段的特征是:硬件不具备通用性,软件更不具备通用性。

② 软件时代(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。在这个阶段中,软件开始作为独立的产品在市场上出售,并能满足不同用户的需要。软件的功能日益丰富、规模日益扩大、结构日益复杂。软件的开发开始以“程序员小组”或“程序员群体”的协作方式进行。60年代末期,IBM公司实行价格分离政策,软件第一次独立于硬件单独定价。这项政策的执行大大促进了软件的发展。几万条、十几万条、百万条指令的软件开始出现,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,尤其在软件规模扩大、复杂性增加以后,软件的开发成本以及完成时间的控制比较困难。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能控制。60年代中期爆发了严重的软件危机。危机的发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。

③ 软件工程时代(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,正式提出软件工程的概念,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。

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,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 软件产业化时代(90年代以后)。软件的工程化加快了软件产业化的步伐。进入80年代中期以来,微型计算机的异军突起,从更大的范围加快了全社会的信息化进程。信息处理过程的计算机化更加突出了软件在信息化发展中的重要作用。日本首次提出了软件立国的主张。IBM公司也打出了“软件推动世界”的广告词。软件产业化发展,使得软件从传统的技术性应用,大规模地向消费性应用过渡,从技术驱动型的发展模式向应用消费驱动型的发展模式转换,这就为软件的产业化创造了有利条件。这又必将大大拓展软件的应用领域,使软件产业成为信息产业的核心主体产业。今天,软件已经主导着信息产品的开发和信息技术市场的开拓,软件已名符其实地成为国民经济信息化和社会信息化的战略性产业。

1.2 计算机系统工程

计算机系统工程的思想最初源于问题求解的思维活动。随着这种求解活动的模型化,理论上建立了相应的计算模型。而微电子、数字技术的发展,逐步将抽象化的计算模型予以逻辑

上、物理上的实现。计算机的发明,标志着计算机系统工程的建设。

一个计算机系统,从信息处理的层面上可以分为硬件系统、软件系统。硬件系统的功能和任务是为软件的运行提供执行环境,同时要为软件提供存在的载体。软件系统的功能和任务是实现信息的获取、分析、处理及输出用户期望的结果。

1.2.1 硬件系统及应用

分立的半导体元器件提供了信号及数字处理的基本功能。硬件设计的任务就是,将分立的半导体元器件以某种电路组合的形式装配成实现一组特定功能的印制电路板。各电路板之间相互用电缆线连接构成系统部件。系统部件采用总线(bus)综合成为特定用途的硬件系统。

硬件工程的设计原理及方法几十年来有了较大的进步,大大促进了半导体工业的发展。今天的半导体工业已经能为计算机系统提供大量功能强大、性能较高、成本较低、使用方便的电子元器件、电路芯片,以及通用性、兼容性较好的电路板或功能全面的系统部件。利用接口及集成技术,计算机应用人员能较容易地为用户构建各种各样的硬件系统。

硬件系统作为计算机应用中的运行支持平台,目前主要有以下几方面的应用:

1) 信号、信息的基本处理

软件的表达形式是信息,而信号与信息的获取、传输、存储、判断及输出等工作必须得到硬件的支持。因此硬件系统的基本功能就是实现信息物理级别的控制与处理。

2) 过程控制及实时控制的应用

工业化生产及控制过程中,需要采集、产生、输出大量的模拟信号及生产数据,这些信号的转换工作与数据的实时控制处理同样需要硬件系统的支持。

3) 嵌入智能

计算机最基本的功能就是信息处理。生活中许多有形实体中存在着大量的信息需要处理。如果能将计算机“嵌入”到这些有形实体中,等于为这些有形的实体安装了能及时处理信息的“大脑”。嵌入的计算机可以代替人处理这些实体内部的信息,实现有形实体工作的自动化。从这个意义上,我们就说该设备有了智能。这就叫嵌入智能。

1.2.2 软件系统及其应用

在今天的计算机系统中,绝大部分的功能均由软件实现。可以说,软件系统是计算机系统的核心和灵魂。

软件的外在形式是信息(主要是指指令和数据),因此,软件有两种基本形态:可执行代码和不可执行代码。

可执行代码是由能在硬件上执行的机器指令组成的序列构成,也可以是由程序设计语言中的可执行语句的序列构成。不可执行代码属于软件的定义、说明范畴,主要由伪指令、说明性语句和数据信息组成。

程序设计语言中提供了多种类型的语句,这些语句实现了各自的基本功能。设计人员按求解问题的需要设计相应的求解算法,并使用这些语句组合成为实现算法的源程序。源程序通过编译、连接,转换成计算机可执行的目标程序,再通过组装多个目标程序构成可执行的软件系统。如图1.1所示。

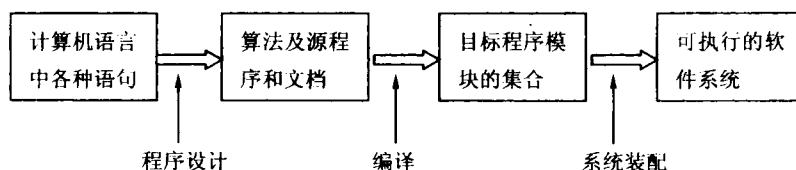


图 1.1 软件系统的构造流程

目前计算机软件主要应用于以下几个方面：

1) 科学计算、数值分析、数据拟合等方面的数值应用

数值计算是计算机软件早期应用最广的领域。今天它仍然在计算数学、空间技术、原子能、数值天气预报、大地测量及工程力学计算等领域中有更高层次的应用。

2) 数据信息的分析、处理

软件从数值计算领域拓展到数据(信息)处理领域,可以说是计算机应用一次划时代的飞跃。正是由于软件进入了信息处理的时代,才有了今天计算机应用的巨大成就。

3) 生产过程及实时控制信息的分析、处理、仿真应用

工业化生产过程中会产生大量的静态数据和实时动态数据,软件对这些数据的实时收集、处理、仿真、输出的应用过程,构成了工业自动化的基础。

4) 电子游戏、视听等娱乐应用

软件在娱乐方面的应用,已使软件从纯技术性产品向消费性产品过渡,如游戏软件、影视动画产品等。

5) 多媒体信息的共享、查询等服务性应用

信息的社会化,社会的信息化是计算机软件在信息增值服务领域应用的自然结果。它将改变人类在信息世界中的生存和驾驭能力。

6) 信息通信及网络应用

软件与通信的结合,无疑是当今最热门的应用领域。网络的应用、通信的便利将改变人类的生活方式和影响社会文明的发展进程。

1.3 软件的特征

软件这个概念,从它出现之时,就带有一层神秘的色彩。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。那么什么是软件呢?早期人们对软件的定义是:用来完成某些任务的程序和数据的集合。程序是指令的序列,指令是能在计算机硬件上执行的动作。而数据是指令操作的对象。随着技术的发展,人们对软件的认识也在加深。今天,用软件工程的观点看,软件的定义应当是:软件是完成某类问题求解的程序和数据以及为维护程序必须提供的一系列文档组成的集合。用简洁的公式可表示为:

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

1.3.1 软件的本质

事物的本质是通过内外两个方面来表现的。软件作为概念世界中一个重要的概念实体,

我们也得从其内部性质谈起。

① 软件的内部性质是:软件具有高度的抽象性和严密的逻辑性。

软件是问题求解方法的信息表达形式,而问题求解方法(计算机中称为算法)是高度抽象的思维活动。因此,高度的抽象性是软件与生俱来的本性,人们无法直接感知软件,必须通过认识、理解、判断、推理等一系列复杂的思维过程才能感知它、认识它、理解它,从而得到它。

软件是大量逻辑元素的复杂组合。这些逻辑元素可以是变量、数组、记录、文件、标号、常数等数据结构,也可以是循环、转移、条件、顺序、推理、赋值等控制机制,甚至还可以是环境、人、其他软件、硬件等外部元素。因此软件中涉及的逻辑量比硬件系统要多出10~100倍。为完成一个复杂的大型软件,常常需要建立一个庞大的逻辑体系。严密的逻辑性是指这个复杂的逻辑体系中,各种逻辑元素之间的联系必须是一致的、无矛盾的,体系结构及其表示必须是统一的。任何逻辑联系上的失误都将导致软件的错误,严重的将导致软件的失败。

② 软件的外部性质是指:软件是一种逻辑的信息产品,是用文字、符号表达的智力产物。

软件的内部性质决定了软件的外部形式只能是逻辑的、思维的结果,其外在表达形式只能通过抽象的符号表达。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。

1.3.2 软件产品的特征

由于软件本身具有的特殊性质,使得软件产品具有以下特征:

① 软件是知识、技术、资金高度密集的产品。

软件是思维产物的本性决定了软件的开发成本主要表现在智力、知识和技术方面。当今社会,科学技术的价值已经得到了社会的广泛承认,而作为科学技术的创造者——人的价值更应得到重视。一个软件项目的成功,很大程度上取决于开发人员的智力和技术水平。而一个大型的软件项目,又需要集中一大批高素质的技术人才的智慧,合理分工,协调配合。例如,IBM公司60年代开发OS/360操作系统,前后共集中了几百人参与开发,花费了2亿美元。美国苹果计算机公司(Apple)仅开发一套家用软件系统,就耗资5000多万美元。软件产品的这一特征启示我们,在软件开发过程中要尊重知识,尊重人才,同时还要求管理中必须精密规划,严格成本核算,控制开发成本。

② 软件产品无明显的生产制造过程,一旦开发成功,复制十分容易。

这一特征也可以从软件的本质中找到依据。软件既然是思维活动的产物,那么它的分析、设计的过程及产品的雏形都隐含在开发人员的头脑中。最后以抽象符号体系表达的软件,只是这个思维过程的结果。这与工业生产中产品的制造过程是完全不同的。既然软件是用符号表达,而符号的复制又十分容易,这就导致软件的复制非常容易。这个复制过程非常类似于复印机的复印过程。软件产品的这一特性启示我们,软件开发过程的管理应不同于一般产品的生产过程管理。必须采用特殊的管理技术与方法,加强过程及中间结果的复审。同时还必须加强软件的知识产权保护,以法律手段维护软件投资者或开发者的合法权益。

③ 软件永不会磨损、变旧,只会过时与失效。软件需要不断地“进化”升级。

软件是用符号表达的逻辑产品,它不具备任何物理特征。平时它以静态的0,1二进制形式存储于介质上。当在计算机上运行时,它又以电信号的形式游历于计算机内各个逻辑部件中,完成赋予它的使命。既然软件没有物理特征,当然也就不会磨损、变旧。存储软件的介质

可能会损坏,但软件是不会损坏的。只要软件有备份,换个介质存储,软件照样可以执行。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。失效是指软件信息有意无意地被破坏(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我们,应不断地适应用户市场的需求变化,以及运行环境的改变,对软件进行完善、扩充,促使软件的升级,以免软件过早地过时而被市场淘汰。

④ 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。

符号的最大特性是可变性和流动性,我们可以随心所欲地增加、删除、修改、移动和传输符号。软件既然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、插入、删除软件的各个部分。这一特性为软件的组织 and 构造提供了一条可行的途径。目前广泛流行的面向对象方法以及即插即用的“软构件”生成方法就是利用了软件产品的这一特性。

⑤ 软件具有强烈的个人色彩,体现了开发者的个人风格。

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。软件产品体现着设计人员的思维特点和习惯做法(统称为个人风格)。我们将软件的这一特性称为软件的属人性。从工程化的角度看,软件的这一特性弊大于利。软件不是艺术品,它需要交流,需要不断地维护、升级。而要做到这一点,就要求软件结构清晰,容易理解。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,以提高软件的可读性和可维护性。

⑥ 软件产品的开发是程序和相关文档的开发。

软件作为一种特殊商品,其功能和内容应不断地升级,以适应用户对软件不断增长着的需求。软件的升级过程需要修改内部源代码,而这项工作要求对软件原有的内容及结构有深入的了解。文档就是详细记录在软件开发过程中设计人员约定的准则、设计的思想、实现的细节等方面内容的文、图、表资料。借助相关的计划、分析、设计及实现的文档,软件维护人员才可能深入了解软件的内部结构,才能够对软件进行必要的修改和扩充,保证软件产品的不断升级。

1.4 软件危机

什么是软件危机?软件危机就是软件在开发过程中存在的一系列严重问题的总称。在软件的发展进程中,软件危机的出现具有两方面的积极意义:一方面由于软件危机的爆发,引发了软件行业的一场灾难,许多软件项目无法正常运行,甚至失败,带来了经济上、技术上、心理上的损失;另一方面,软件危机也促使人们从深层次上反思软件的本质及开发技术、方法上存在的严重缺陷,促使计算机领域发展新技术,寻找解决软件危机的途径。

软件危机中所指的一系列严重问题是指:

① 软件的复杂性日益增加,因而软件的开发成本及进度无法正确预测,导致实际成本大大超过预算成本,开发风险增大。

早期开发的软件,由于规模较小,复杂性不高,因而软件开发成本、时间和进度较容易控制。随着软件规模的扩大,复杂性随之增加,几万条、几十万条、几百上千万条指令的软件系统相继出现。如1965年美国航空公司使用的SABRE订票系统,规模达38万条指令;IBM公司

1965年推出的OS/360操作系统,规模达47万条汇编指令;随后推出的美国阿波罗登月飞行的控制软件有1000多万行指令;航天飞机的控制软件更是高达4000多万行。这些软件系统的复杂性早已超过传统的开发、管理技术所能驾驭的程度。面对如此规模的大型软件系统,其开发存在的风险是非常巨大的,一旦开发不成功,损失难以承受。

② 软件产品的质量无法保证,可靠性较低,难以满足用户需求。

传统的软件开发方法,人们过于重视算法的精巧和效率,将软件开发视为一种显示个人能力的技巧而忽视了软件的质量和今后的维护。软件是一种极度精密的系统,其分析、设计、编程中的失误都将导致软件质量的下降。而传统的软件设计方法中又缺乏相应的质量保证手段,开发人员的经验、素质甚至情绪都将会影响软件的质量。质量差的软件要么运行效率下降,运行成本增加,影响用户的业务发展,要么无法正常运行,导致投资失败。

③ 交付使用的软件无法维护,难于移植、兼容,以至于不得不重复开发类似的软件。

人们在软件的使用上存在一个误区,认为开发的软件只要能在计算机上运行就可以了。殊不知能运行只是软件必须满足的基本条件。软件不是一次开发成功就终生受益的,而是需要不断地修改、完善、扩充,不断地适应环境、技术、需求的变化,或者说软件需要不断地升级,才能生存,才能获得市场。

④ 传统的手工生产方式使软件生产率较低,难以满足社会对软件日益增长的需要。

硬件技术的发展创造了一个微电子学的奇迹:性能越高的器件,价格反而更便宜。这是生产率提高以后,规模效益产生的结果。而软件由于一直采用手工生产方式,这对于早期规模较小的软件来说还算合适,但对于大型软件项目来说,这种生产方式就显得力不从心。虽然后期人们采用程序员小组或软件组织的方式协作生产软件取得了一定的效果,但软件的生产方式本质上仍无法摆脱个人作坊的方式。多人协作共同开发的方式,在当时技术环境的条件下,如果没有一套规范的过程管理方法,没有统一的信息交流工具,没有一定强制性的约束规则,不仅不能提高软件生产率,反而可能使开发成本增加,失败的风险加大。

产生软件危机的原因是多方面的,有观念上、技术上、环境上及人为方面的因素。但技术上的原因主要有:

① 软件的设计方法存在致命的缺陷。

在软件的生产过程中,想要“看见”进展相当困难。因此软件项目的完成,只能不断地重复“修改—测试—再修改—再测试”的迭代过程。在一个迭代过程何时收敛的问题未解决之前,软件开发的成本及进度是无法预先控制的。

② 软件的逻辑本性以及其构成的庞大复杂的逻辑体系使开发人员感到束手无策,难以驾驭。

软件是由众多逻辑量以错综复杂的联系构成的,这些联系中只要有一种联系不正确,都将影响软件整体上的 consistency、正确性。

③ 软件是逻辑的而不是物理的,因此软件的质量无法根据各组成部件的质量来衡量。对软件的任何一个局部改动都必须从软件整体上进行校正和测试。这就给软件的质量保证带来困难。

④ 实现软件的工具——程序设计语言本身存在严重的缺陷。

早期软件的实现工具只有为数不多的几种程序设计语言。这些语言在结构上、表达能力上存在严重的问题,使得软件的逻辑结构不清楚,数据格式不统一,可读性极差,难以相互交流

或者互审,这给下一步软件的修改、测试造成困难。

⑤ 软件的设计方法与人类认识问题的思维方法之间存在差异,或者说问题的求解方法的空间结构与问题的认识分析的空间结构不一致。这种差异或不一致在缺乏有效的技术手段予以弥补或缓和时,表现得更为突出和严重。

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。软件危机不会在短时期内消除,而人类除了解决软件危机这条路之外别无它途。多年来人们一直在努力研究寻找解决软件危机的途径。从目前的发展结果看,解决软件危机的途径主要有以下 3 种:

① 研制新一代体系结构的智能型计算机,改变软件实现的方式,以此降低软件的复杂性。

使用 Von. Neuman 体系结构的计算机,如果用于求解数值运算类的问题,是非常合适的。但用于求解非数值类问题(即:符号的处理问题),就必须用过程性的程序设计语言详尽描述其求解的算法过程,然后再转换为一系列简单的操作指令和逻辑判断(Von. Neuman 体系结构的计算机只会做这些简单操作),这个过程就使得软件非常复杂。

如果能对计算机硬件体系结构进行革新,人们无须详尽描述问题的求解过程,而只要给出问题的描述和求解的规则,求解过程由计算机自动实现。这就可以从根本上大大降低软件设计与实现过程的复杂性,从而消除软件危机。到目前为止,具有这种能力的新一代计算机(也称为第五代计算机)还未研制成功,而相关的理论及方法已有实际应用成果,但仍然要借助于传统的计算机硬件实现。如现在流行的逻辑程序设计方法、第四代程序设计语言(4GL)等。

② 软件工程的解决途径。

应用一套工程化、规范化的系统方法来指导软件的开发、运行、维护过程,探索研究更好的软件设计、表达技术及管理方法,尽可能地使用软件工具及软件工程环境辅助软件的生产,保证软件的质量。这就是软件工程途径的基本内容。软件工程是 70 年代以来化解软件危机最成功,并在实用性、经济性方面取得较好效果的主要的途径,也是本书要讲授的主要内容。

③ 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。

软件工程途径面临的一个困境是:用于软件设计的方法空间同实际客观世界的问题空间存在差异,从而产生了较严重的映射误差。90 年代以后逐渐流行的面向对象的软件设计方法,统一了思维空间和方法空间的结构,消除了软件设计中存在的映射误差,从方法学上改善了软件的设计过程。面向对象方法必将成为软件设计的主流方法。本书第 8 章将重点介绍面向对象的软件设计方法。

1.5 软件工程的观念

“软件工程”这个概念是在 1968 年北大西洋公约组织(NATO)主持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指导软件的开发、运行及维护工作。

1.5.1 软件工程的定义

什么是软件工程?不同的机构从不同的角度给出了许多定义。如,IEEE 软件工程专业术语中给出的定义是“开发、运行、维护软件以及使之退役的一套系统方法”。

我们认为比较规范的定义应该是:软件工程是一门研究如何将完善的科学与工程原理,先进高效的管理技术以及完整、规范的设计方法应用于经济地开发既可靠,又能在实际计算机上有效运行的软件的一门工程学科。

在以上定义中,“完善的科学与工程原理…”的含义是,软件工程将应用计算机科学、数学、工程管理科学的原理及借鉴其他工程领域的原则、方法和成功经验,建立并发展一套适用于软件工程的规范化方法。其中计算机科学以及数学用于软件模型的构造及算法设计;工程学用于制订软件工程的规范及文档标准,并建立一整套用于软件成本分析、监督复审的技术;管理科学则用于软件的计划、资源、质量、成本的控制与管理。

“经济地开发”的含义是,软件开发成本的预算能控制在合理的范围之内。由于软件是资金、人力高度密集的产品,成本控制方面的失误或开发周期的延长,都将大大增加软件的开发成本,若控制失当将导致项目的失败。

“既可靠,又能有效运行”的含义是,软件质量符合质量要求,软件不仅能正确运行,并且无故障运行的时间能达到要求的标准。

1.5.2 软件工程的基本原则

作为一门专门研究软件开发、运行、维护的工程学科,软件工程追求的目标是:需求上的可用性、实现上的正确性、经济上的合算性。为实现这个目标,围绕着软件的分析、设计及管理,软件工程确定了几条基本原则,这些基本原则一直作为软件工程学科发展的指导方针。

① 软件开发的各个阶段应当采用较好的软件表示形式及规范的文档格式。

传统的软件开发过程是一个“隐含的思维过程”,人们想要看到这个过程是很困难的。软件工程要求将软件开发过程划分为若干个阶段,每个阶段必须采用较好的软件表达工具表示软件的开发进展,每个阶段又必须以文档的形式记录下软件分析、设计过程中的有关信息,便于专门机构对软件开发的进展实施有效的复审。

② 软件在实现过程中应设法提高软件系统部件的可复用性。

对于计算机硬件,我们要求其通用性、兼容性较好,这样做的目的是为硬件设计人员提供较大的选择余地,便于按设计要求集成硬件系统。而对于软件,软件工程中也要提高部件的可复用性,同样也便于软件设计人员以组合集成的方式开发软件系统。

③ 软件设计应大力研究与发展自动设计技术,使软件的生成自动化。

软件是人类智力的产物,而人类的思维具有从事低级、重复性、机械性的工作时极易出错的特点。为保证软件产品的质量,降低设计过程中出现错误的概率,开发研制一些软件生成工具自动生成软件中的某些部件,无论从技术上、经济上都是合算的。

④ 必须研究建立一套协调、控制并管理软件开发过程的理论、技术及方法。

软件的开发过程是一个技术相互交叉、人际相互交往、信息相互交流的复杂的系统工程。这其中,技术因素、人的素质因素、经济因素、环境因素等都可能成为软件开发失败的原因。要保证这些因素的协调、可控,就必须建立一套工程管理的规范方法,从而建立软件工程的新秩序。

序。

1.5.3 软件工程的研究内容

了解了软件工程的定义及指导原则后,对软件工程这门学科的研究内容就很容易理解了。软件工程的研究内容见图 1.2。

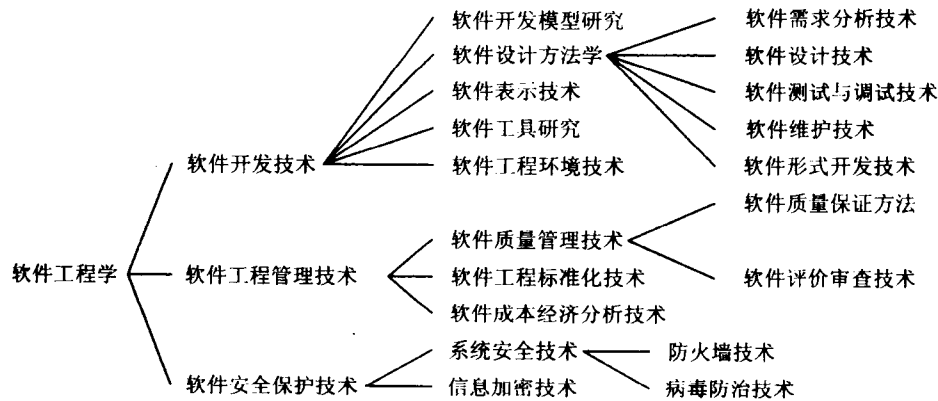


图 1.2 软件工程的研究内容

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,它研究的内容涉及面较广。其概念、原理、方法及管理的思想广泛来源于计算机科学、数学、工程学、管理科学、成本经济学、心理学及运筹统计学等学科。

2) 方法的实用性

工程的概念就意味着软件工程学科不是以理论研究为目的,而是倡导用一套工程化、实用化的方法指导或实践软件的开发活动。

3) 内容变化的动态性

软件工程学科的研究内容不是静态不变的,它是随着理论与技术的研究与发展,不断淘汰过时的内容而充实与发展新的技术、新的概念、新的方法。其内容的动态变化是与技术的发展同步的。

1.5.4 软件工程的研究途径

软件工程的目标是明确的。但在如何实现目标的问题上却存在 2 种不同的研究途径。

1) 工程学派

这一学派的基本思路是,应用工程化的原理和方法建立软件的工程化模型,并采用先进的开发方法和科学的工程管理技术指导、控制、复审软件的开发及维护过程。这种工程化的研究途径从思想上、方法上丰富了软件工程的内容,并在软件工具的研制技术、集成技术及软件工程环境方面有了较大的发展。现已成为软件工程研究的主流。本书主要介绍的就是工程学派的软件 engineering 内容。

2) 形式学派