

WUTP

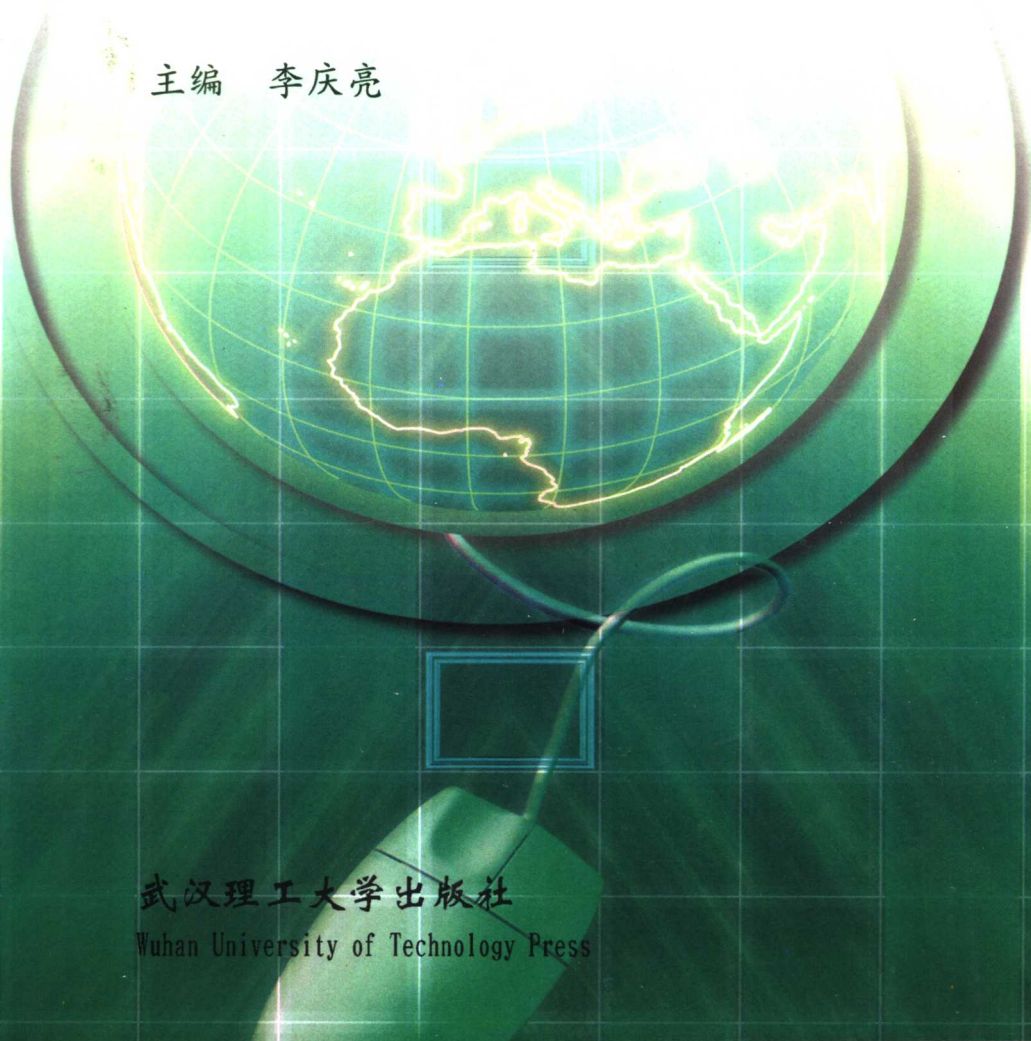
面向21世纪
高职高专计算机类
专业新编系列教材

Software
Engineering

软件工程

主编 李庆亮

武汉理工大学出版社
Wuhan University of Technology Press



面向 21 世纪高职高专计算机类专业新编系列教材

Software Engineering

软 件 工 程

主 编 李庆亮

副主编 刘德清 崔晓军

武汉理工大学出版社

Wuhan University of Technology Press

内 容 简 介

本书以软件的生命周期作为主线索,重点讨论了结构化的软件开发方法和技术,包括需求分析、系统设计、编码、测试和维护等。随着近年来面向对象软件开发方法和技术的研究及应用不断普及,本书也利用一定篇幅介绍了面向对象的分析和设计方法。

软件工程是一门实践性很强的学科,书中内容尽量做到通俗易懂,原理与实例相结合,强调实用性。

本书读者对象是高职高专层次的计算机专业学生,也可以供从事计算机软件开发与应用的工程技术人员作参考书。

图书在版编目(CIP)数据

软件工程/李庆亮主编. —武汉:武汉理工大学出版社,2004. 8
(面向 21 世纪高职高专计算机类专业新编系列教材)
ISBN 7-5629-2122-9

I. 软... II. 李... III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字(2004)第 051483 号

出版发行:武汉理工大学出版社(武汉市洪山区珞狮路 122 号 邮编 430070)

HTTP://www.techbook.com.cn

E-mail:duanchao@mail.whut.edu.cn tiandq@mail.whut.edu.cn

经 销 者:各地新华书店

印 刷 者:安陆市鼎鑫印务有限责任公司

开 本:787×960 1/16

印 张:13

字 数:253 千字

版 次:2004 年 8 月第 1 版

印 次:2004 年 8 月第 1 次印刷

印 数:1—5000 册

定 价:18.00 元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。本社购书热线电话:(027)87397097 87394412

凡使用本教材的教师,可拨打(027)87385610 免费索取电子教案光盘。

出版说明

面向新世纪,我国高等职业技术教育进入蓬勃发展的新时期。根据 IT 行业技术新、发展快的特点,高等专科学校、高等职业技术学院计算机类专业教育,按照社会主义市场经济规律的原则定位人才培养目标和调整教学方法,尽量按照新技术或新版本更新课程内容,加速各种新产品和新技术的推广应用,努力提升高等职业技术教育对国民经济发展的促进作用。

根据高等职业技术教育快速发展与教学改革对教材建设的需求,武汉理工大学出版社经过广泛调研,与国内近 30 所高等专科学校、高等职业技术学院的计算机教育专家进行探讨,决定组织编写一套适合于高等职业技术教育计算机类专业(涵盖计算机应用与维护、计算机网络技术、计算机软件技术等专业方向)人才培养和教学需要的具有特色的高质量教材——面向 21 世纪高职高专计算机类专业新编系列教材。

本套新编系列教材的编写具有以下特色:

1. 与时俱进,教材内容体现人才培养目标

本套教材的编写反映教育部制订的《高职高专教育基础课程教学基本要求》和《高职高专教育专业人才培养目标及规格》的文件精神,贯彻高等职业技术教育“要服务于社会主义现代化建设,要与生产劳动和社会实践相结合”的宗旨,以培养一大批满足生产第一线需要的高等技术应用型人才为目标,坚持以技术应用型为主线的原则来编写教材内容,加强应用能力的培养。

2. 紧跟教学改革步伐,体现教学改革阶段性成果

本套教材的编写反映高职高专学校教学改革的阶段性成果,在处理“基础理论”与“实践能力”之间的关系上,遵循“基础理论以够用、必需为度,突出应用”的原则。教材编写坚持“少而精”的原则,以培养从

事计算机应用与维护、网络建设与维护及软件开发与测试等方面的能力,并能够快速跟踪计算机新技术发展的高等技术应用型人才为目标。坚持理论与实际相结合,采用“提出问题—分析问题—设计任务—解决任务—总结规律”的编写方法,努力创造出高职高专教材新体系。

3. 实现立体化出版,适应教育方式的变革

本套教材努力使用和推广现代化的教学手段,凡有条件的课程都准备组织编写、制作和出版与教材配套使用的实验、习题、课件、电子教案及相应的程序设计素材库。

本套教材首批 26 种预计在 2004 年秋季至 2005 年春季全部出齐。我们的编审者、出版者决不敢稍有懈怠,一定高度重视,兢兢业业,按最高的质量标准工作。教材建设是我们共同的事业和追求,也是我们的共同的责任和义务,我们诚恳地希望大家积极选用本套教材,并在使用过程中给我们多提意见和建议,以便我们不断修订、完善全套教材。

武汉理工大学出版社

2004 年 1 月

面向 21 世纪高职高专计算机类专业 新编系列教材编审委员会

顾问:

钟 珞 危道军

主任委员:

舒云星 雷绍锋

副主任委员:(以姓氏笔画为序)

刘德清 李庆亮 张树臣 张浩军 周松林

郭长庚 徐卓峰 崔轩辉 常荆燕 黄春喜

委员:(以姓氏笔画为序)

丁文华 王一兵 王学军 王海芳 刘自强

孙清伟 宋锦河 李京秀 李晓桓 何月顺

陈 年 陈松才 陈桂生 陈 鑫 张有谊

张晓云 张新成 苏 玉 周 舸 金 平

武 新 欧晓鸥 赵丽梅 赵 静 姜华斌

徐立新 徐善荣 秦振吉 郭荣冰 黄亚平

崔晓军 戴春霞

秘书长:田道全

总责任编辑:段 超 徐秋林

前 言

为解决软件生产率和软件质量远远满足不了社会对软件数量、功能和质量等方面的需求这一软件危机问题,自从 1969 年提出软件工程的观念以来,历经 30 多年的探索和发展,软件工程的理论和技术已逐渐成熟,现已成为计算机科学和技术领域的一个重要分支。目前,软件工程所涉及的范围包括软件开发技术、软件工程环境、软件经济学、软件心理学和软件工程管理等方面知识。

由于软件工程是一门实践性很强的学科,只有在不断的学习和实践过程中认真体会和理解,才能深刻掌握软件工程的实质和内涵。因此,本书理论和实用并重,在讲述软件工程的基本原理、概念、方法和技术的同时,重点讲述软件工程在实践环节中的应用。

本书特色如下:

(1)由于结构化的软件开发方法和技术是初学者入门的基础方法和实用技术,因此本书以软件的生命周期作为主线索,重点讨论了需求分析、系统设计、软件编码、软件测试和系统维护的基本概念、原理、技术和方法。内容尽量做到通俗易懂,图文并茂,原理、方法与实例相结合。

(2)面向对象方法符合人们认识客观世界、解决复杂问题的渐进过程,是当今比较流行的软件开发方法之一。随着近年来面向对象软件开发方法和技术的研究及应用不断普及,本书也用一定篇幅介绍了面向对象的系统分析和设计过程及面向对象的实现,每个环节都给出了具体的步骤。

(3)软件编码是计算机专业学生的课程学习和课程实践中的重要内容,也是专科层次学生毕业后的主要工作岗位。编码的风格在很大程度上决定着程序的质量和效率,本书在软件编码部分采用了大量实例,重点讨论了影响编码风格的各种因素,希望读者能在日常的程序设计和软件开发实践中养成规范的编码习惯,不断积累经验。

(4)为使初学者能够对软件开发的方法、技术和过程在实际开发实践中的应用有一个整体的把握,并在学习和实践过程中得到软件工程方法和技术的实际训练,本书最后用一个较完整的实例讲述了小项目型软件开发的一般方法和过程,并结合实际提供了大量实训题目,希望能为读者提供一定的帮助。

本书各章后都附有内容小结和习题,便于学生课后复习和掌握所学内容。

本书可作为高职高专计算机专业的教材,也可以供从事计算机软件开发与应用的工程技术人员作参考书。

参加本书编写工作的有:河南机电专科学校李庆亮、李吉彪、马绍惠;襄樊职业技术学院刘德清、崔晓军;许昌职业技术学院连智锋;南京交通职业技术学院高水娟;湖北生态职业技术学院王广胜。全书由李庆亮统稿主编。

由于时间仓促,加之水平有限,书中难免存在不足之处,恳请广大读者和专家批评指正!

编 者

2004 年 5 月

目 录

1 绪论	(1)
1.1 软件与软件危机	(1)
1.1.1 软件的特点	(1)
1.1.2 软件生产的发展	(2)
1.1.3 软件危机	(2)
1.2 软件工程	(4)
1.2.1 软件工程的定义	(4)
1.2.2 软件工程的内容和目标	(5)
1.2.3 软件工程的基本原则	(5)
1.3 软件工程面临的主要问题	(6)
1.4 软件开发方法	(7)
1.4.1 结构化开发方法	(7)
1.4.2 Jackson 方法	(7)
1.4.3 面向对象的开发方法	(8)
1.5 软件工程的学习	(8)
习题与思考题	(10)
2 软件生存周期与软件开发模型	(12)
2.1 软件生存周期	(13)
2.2 软件开发模型	(15)
2.2.1 瀑布模型	(16)
2.2.2 快速原型模型	(18)
2.2.3 增量模型	(18)
2.2.4 螺旋模型	(20)
2.3 可行性研究	(21)
2.3.1 可行性研究的任务	(21)
2.3.2 可行性研究的步骤	(22)
习题与思考题	(24)

3 需求分析	(26)
3.1 需求分析的特点	(26)
3.2 需求分析的任务与步骤	(27)
3.3 需求获取的内容、方法与原则	(29)
3.3.1 需求获取的内容	(29)
3.3.2 需求获取的方法	(30)
3.3.3 需求分析的原则	(31)
3.4 结构化分析建模	(35)
3.4.1 结构化分析模型	(35)
3.4.2 结构化分析与描述工具	(35)
3.5 需求复审	(40)
习题与思考题	(41)
4 软件设计	(44)
4.1 软件设计的基本概念和原理	(45)
4.1.1 模块与模块化	(45)
4.1.2 抽象与细化	(47)
4.1.3 信息隐蔽	(48)
4.1.4 模块的独立性	(48)
4.1.5 由底向上与自顶向下设计	(52)
4.1.6 结构化设计原则	(53)
4.2 概要设计(总体设计)	(55)
4.2.1 从分析模型导出设计模型	(56)
4.2.2 面向数据流的设计方法	(56)
4.2.3 变换设计	(60)
4.2.4 事务设计	(64)
4.2.5 设计的后处理	(65)
4.3 详细设计	(66)
4.3.1 详细设计的目的与任务	(66)
4.3.2 详细设计的原则与方法	(67)
4.3.3 详细设计工具	(70)
4.4 面向数据结构的设计方法	(75)
4.4.1 Jackson 表示法	(75)
4.4.2 Jackson 程序设计方法	(76)
习题与思考题	(81)

5 编码	(83)
5.1 程序设计语言的特性及选择.....	(83)
5.1.1 程序设计语言特性	(83)
5.1.2 编码语言的选择	(85)
5.2 编码风格.....	(86)
5.2.1 正文编排格式	(87)
5.2.2 注释	(88)
5.2.3 标识符命名	(90)
5.2.4 可读性	(91)
5.2.5 变量、结构	(94)
5.2.6 效率	(95)
5.2.7 其他要注意的问题	(99)
习题与思考题.....	(100)
 6 软件测试	(102)
6.1 软件测试的基本概念	(102)
6.1.1 软件测试的目的与任务	(102)
6.1.2 软件测试的特性	(103)
6.1.3 软件测试的基本原则	(105)
6.2 软件测试方法	(105)
6.3 测试用例的设计	(106)
6.3.1 逻辑覆盖	(107)
6.3.2 等价类划分	(110)
6.3.4 边界值分析	(112)
6.3.5 错误猜测法	(114)
6.4 软件测试的步骤	(114)
6.4.1 单元测试	(114)
6.4.2 集成测试	(118)
6.4.3 确认测试	(120)
6.4.4 系统测试	(121)
6.5 纠错技术	(122)
6.5.1 纠错的策略	(122)
6.5.2 常用的纠错技术	(123)
习题与思考题.....	(126)

7 软件维护	(129)
7.1 软件维护的概念	(129)
7.1.1 软件维护的种类	(129)
7.1.2 软件维护的代价	(130)
7.1.3 软件维护的问题	(131)
7.2 软件维护的实施	(132)
7.2.1 维护机构	(132)
7.2.2 维护申请报告	(132)
7.2.3 软件维护过程	(133)
7.2.4 维护复审	(133)
7.3 软件可维护性	(134)
7.3.1 影响可维护性的软件属性	(134)
7.3.2 软件可维护性的度量	(134)
7.3.3 提高软件可维护性的方法	(135)
7.4 维护的副作用	(136)
习题与思考题.....	(137)
 8 面向对象软件开发方法	 (139)
8.1 面向对象方法基础	(139)
8.1.1 传统方法存在的问题	(139)
8.1.2 面向对象方法的基本概念	(140)
8.1.3 面向对象方法的描述工具	(142)
8.2 面向对象的需求分析	(145)
8.2.1 面向对象分析模型	(145)
8.2.2 面向对象分析的一般步骤	(146)
8.3 面向对象设计	(151)
8.3.1 将 OO 分析模型转换到 OO 设计模型	(151)
8.3.2 对象模型技术(OMT)的 OOD 方法	(152)
8.3.3 Yourdon 的 OOD 方法	(154)
习题与思考题.....	(157)
 9 小软件项目的开发	 (159)
9.1 小软件项目的特点及开发中常犯的错误	(159)
9.2 小项目软件的开发过程	(161)

9.2.1 计划和分析	(161)
9.2.2 设计、编码和测试	(162)
9.2.3 维护和管理	(162)
9.3 实例	(163)
9.3.1 问题的提出	(163)
9.3.2 系统规划	(163)
9.3.3 系统分析	(164)
9.3.4 系统设计	(169)
9.3.5 系统实施	(173)
习题与思考题	(174)
附录 软件开发文档编写规范	(176)
一 可行性研究报告	(176)
二 “可行性研究报告”实例	(179)
三 软件需求规格说明书	(183)
四 概要设计说明书	(186)
五 详细设计说明书	(188)
六 测试计划	(190)
七 测试分析报告	(192)
参考文献	(194)

1 绪 论

本章提要

由于计算机硬件技术的进步,为计算机的广泛应用创造了条件,使计算机应用逐步深入到社会生活的各个方面,同时也促使人们对计算机软件的品种、数量、功能和质量等提出了越来越高的要求。然而,软件的规模越大、越复杂,人们的软件开发能力越显得力不从心。这使得软件生产率和软件质量远远满足不了社会发展的需求,成为社会、经济发展的制约因素。于是,人们开始重视软件开发过程、开发方法、开发工具和环境的研究,软件工程应运而生。

本章介绍了软件工程的基本概念,包括软件与软件危机,软件工程的定义、目标、原则和面临的主要问题,软件开发方法等。最后讨论了在软件工程的学习和教学中应注意的一些问题。

1.1 软件与软件危机

1.1.1 软件的特点

所谓软件是指用来控制计算机运行的各种程序,以及用于开发、使用和维护这些程序的数据和相关文档资料的完整集合。软件与硬件一起构成一个完整的计算机系统,它们相互依存,缺一不可。同硬件相比,软件是一种特殊的产品,它具有下列一些特点:

(1)软件是一种逻辑实体,而不是具体的物理实体,因而它具有抽象性。软件产品是脑力劳动的结晶,总是以程序和文档的形式出现,保存在计算机的存储介质上,通过计算机的运行才能体现它的功能和作用。

(2)软件的生产与硬件不同。软件产品的生产主要是研制,在它的开发过程中没有明显的制造过程,其成本主要体现在软件的开发和研制上,软件开发研究完成后,通过复制就产生了大量软件产品。

(3)软件生产目前还未完全摆脱手工开发方式,大部分产品是“定做”的。

(4)软件产品在运行和使用期间,没有硬件那样的机械磨损、老化等问题。

(5)软件开发成本相当昂贵。由于实际问题的复杂性和程序逻辑结构的复杂性,使得软件的复杂性很高,研制工作需要投入大量的、复杂的、高强度的脑力劳动,成本很高。

1.1.2 软件生产的发展

自从第一台计算机诞生以来,就开始了软件的生产,到目前为止,已经过了程序设计、程序系统和软件工程三个时代。

(1)程序设计时代(1946~1956年)

程序设计时代的生产方式是个体手工劳动;使用的工具是机器语言、汇编语言;开发方法是追求编程技巧,追求程序运行效率,使得程序难读、难懂、难修改;硬件特征是价格贵、存储容量小、运行可靠性差;软件特征是不重视程序设计方法的研究。

(2)程序系统时代(1956~1968年)

程序系统时代的生产方式是作坊式的小集团合作生产;生产工具是高级语言;开发方法开始提出了结构化程序设计方法,但开发技术没有实质上的突破,仍旧靠个人技巧;硬件特征是速度、容量及工作可靠性有明显提高,价格降低;软件特征是由于提出了大量的软件开发需求,使得其他行业人员大量转入软件开发行业,由于缺乏训练,开发人员素质很低。开发人员的素质和落后的开发技术不适应规模大、结构复杂的软件的开发需求,因此产生了尖锐的矛盾,导致了软件危机的产生。

(3)软件工程时代(1968年至今)

软件工程时代的生产方式是用工程化的思想、方法和管理方式生产软件。硬件特征是向超高速、大容量、微型化以及网络化方向发展;软件特征是开发技术有了很大进步,使用了数据库技术、开发工具技术、网络技术、分布式技术和面向对象技术等先进技术来开发软件,但仍未能获得突破性进展,没有完全摆脱软件危机。

1.1.3 软件危机

(1)软件危机的产生和表现

“软件危机”一词最早是1968年北大西洋公约组织在德国召开的软件国际

会议上,针对在计算机软件开发和维护过程中所遇到的一系列严重困难提出的。

在软件发展第二阶段的末期,由于计算机硬件技术的进步,计算机运行速度、容量和可靠性有显著的提高,生产成本显著下降,为计算机的广泛应用创造了条件,提出了大量的软件开发需求,一些复杂的、大型的软件开发项目提出来了。但是软件开发技术一直未能满足发展的要求,许多遇到的问题因找不到解决的办法,使问题积累起来,形成了尖锐的矛盾,导致了软件危机的产生。这些矛盾集中起来主要包括:软件开发周期大大超过规定日期;软件开发成本严重超标;软件质量难于保证等几个方面。具体来说,软件危机主要表现在以下几方面:

①完成时间一再拖延。由于对工作量和开发难度估计不足,计划无法按时完成,开发时间一再拖延。

②经费预算经常突破。由于缺乏软件开发的经验和软件开发数据的积累,使得开发经费的预算很难制定。主观盲目制定的预算,执行起来和实际情况有很大差距,使得开发经费一再突破。

③开发的软件不能满足用户要求。开发初期对用户的要求了解不够明确,未能得到明确表达,开发工作开始后,软件人员和用户又不能及时交换意见,使得一些问题不能及时解决,导致开发的软件不能满足用户的要求,开发失败。

④开发的软件可维护性差。开发过程没有统一的、公认的规范,软件开发人员按各自的风格工作,也没有建立完整、规范的文档,发现问题后进行杂乱无章的修改,运行时发现错误很难修改,导致可维护性差。

⑤开发的软件可靠性差。由于在开发过程中,没有确保软件质量的体系和措施,在软件测试时,又没有严格的、充分的、完全的测试,提交给用户的软件质量差,在运行中暴露出大量的问题。这种不可靠的软件,轻者会影响系统正常工作,重者会发生事故,造成生命财产的重大损失。

(2) 软件危机产生的原因

造成上述软件危机的原因主要是由于软件开发技术面临软件规模、软件复杂性和软件生产效率等方面的问题。

①软件的规模越来越大,结构越来越复杂。随着计算机应用的日益广泛,需要开发的软件规模日益庞大,软件结构也日益复杂。例如 1968 年美国航空公司订票系统达到 30 万条指令;1973 年美国阿波罗计划达到 1000 万条指令;Windows 95 有 1000 万行代码,而 Windows 2000 达到 5000 万行代码,用了 3000 多个工程师,几百个小团队。有人曾估计,软件设计与硬件设计相比,其逻辑量要多达 10~100 倍。对于这种规模庞大的软件,其调用关系、接口信息、数据结构非常复杂,复杂程度超过了人所能接受的程度。

②软件开发管理困难。由于软件规模大,结构复杂,又具有无形性,导致管

理困难,进程控制困难,质量控制困难,可靠性无法保证。

③软件开发费用不断增加。软件生产是一种智力劳动,它是资金密集、人力密集的产业,大型软件投入人力多,周期长,费用上升很快。

④软件开发技术落后。在 20 世纪 60 年代,人们注重一些计算机理论问题的研究,如编译原理、操作系统原理、数据库原理、人工智能原理、形式语言理论等,不注重软件开发技术的研究,用户要求的软件复杂性与软件技术能够解决复杂性的能力不相适应,它们之间的差距越来越大。

⑤生产方式落后。软件仍然采用个体手工方式开发,根据个人习惯爱好工作,无章可遵循,无规范可依据,靠言传身教方式工作。

⑥开发工具落后,生产率提高缓慢。软件开发工具过于原始,没有出现高效率的开发工具,因而软件生产率低下。在 1960~1980 年间,计算机硬件的生产由于采用计算机辅助设计、自动生产线等先进工具,使硬件生产率提高了 100 万倍,而软件生产率只提高了 2 倍,相差十分悬殊。

1.2 软件工程

为了克服软件危机,人们从其他产业的工程化生产得到启示,提出要用工程化的思想来开发软件。从此,软件生产进入了软件工程时代。

1.2.1 软件工程的定义

一般来说,软件除了指计算机系统中的所有程序及文档资料外,还包含有另一种学科含义,即指研究、开发、维护以及使用软件所涉及的理论、方法和技术。工程则是将科学理论和知识应用于实践的科学。在了解了“软件”和“工程”两个概念的基础上,软件工程可定义如下:

软件工程是一类求解软件的工程。它应用计算机科学、数学及管理科学等原理,借鉴传统工程的原则和方法,创建软件以达到提高质量、降低成本的目的。软件工程是一门指导计算机软件开发和维护的工程学科。

该定义说明软件工程的主要思想是在软件生产中用工程化的方法代替传统手工方法。工程化的方法借用了传统的工程设计原理的基本思想,采用了若干科学的、现代化的方法技术来开发软件。这种工程化的思想贯穿到需求分析、设计、实现,直到维护的整个过程。

软件工程是涉及到计算机科学、工程科学、系统科学、管理科学、数学等许多科学领域的一门综合性的交叉学科。计算机科学中的研究成果均可用于软件工程,但计算机科学着重于原理和理论,而软件工程则着重于如何建造一个软件系统。软件工程要用工程科学的观点来进行费用估算、制定进度、制定计划和方