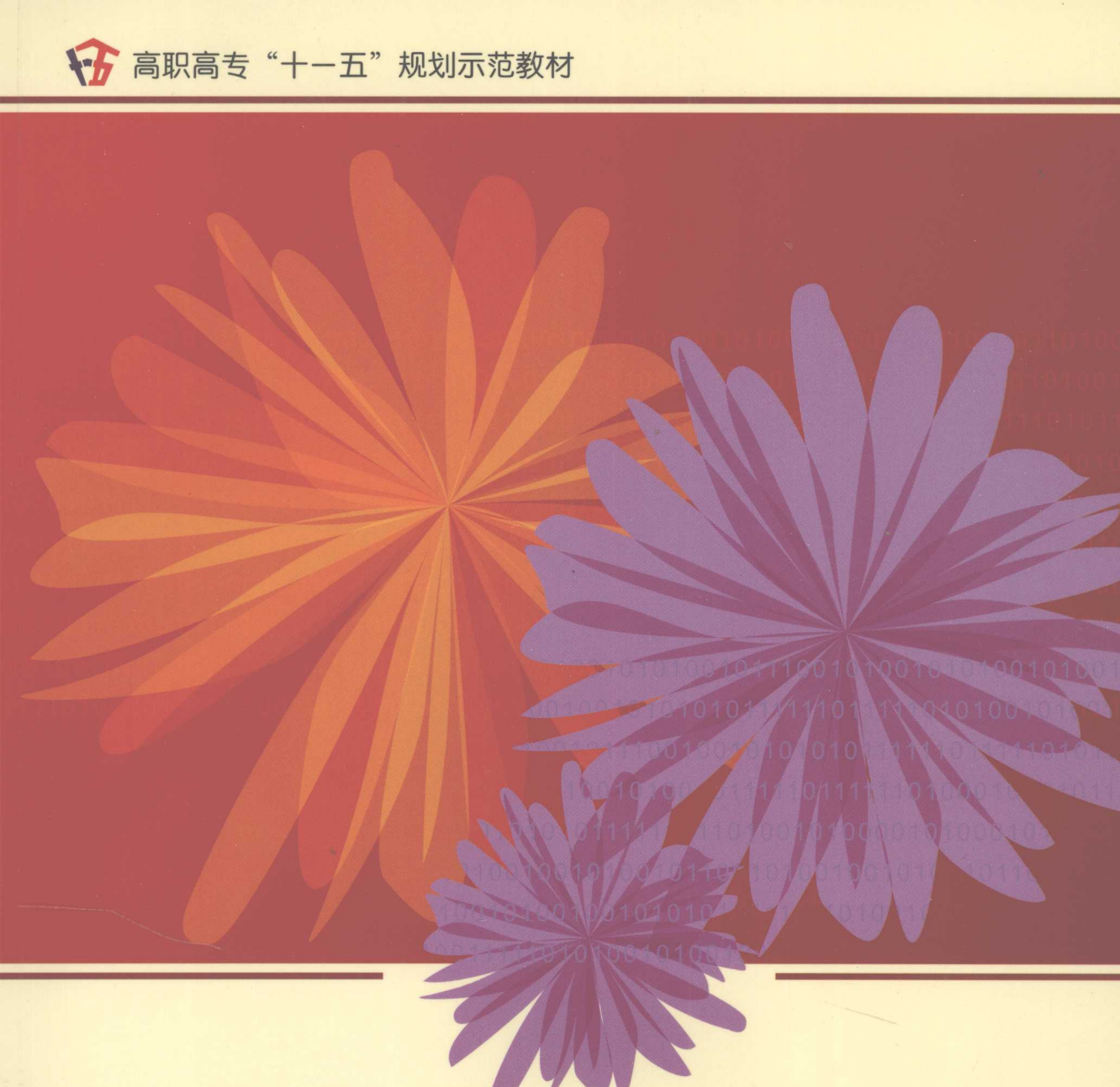




软件工程技术

主 编 张应辉 饶云波
副主编 凌 艳



高职高专“十一五”规划示范教材

软件工程技术

主 编 张应辉 饶云波
副主编 凌 艳

北京航空航天大学出版社

内 容 简 介

本书是根据高职高专教育培养高级技能人才的要求,突出“实践性、实用性、创新性”,并结合作者多年的教学和工程实践经验,以“理论够用、实用为主”为原则编写的一本软件工程技术的教材。内容主要包括:软件工程基础、软件生存周期及开发模型、软件项目开发计划、软件需求、软件设计、软件实现、软件测试和软件维护。

本书参考了大量的最新资料,内容丰富翔实,突出了以实用为原则的特点。

本书既可作为承担国家技能型紧缺人才培养培训的高职高专院校计算机类、信息管理类专业的教材,也可作为网络技术人员的参考书。

图书在版编目(CIP)数据

软件工程技术/张应辉等主编. —北京:北京航空航天大学出版社,2006.8

ISBN 7-81077-800-5

I. 软… II. 张… III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字(2006)第 084658 号

软件工程技术

主 编 张应辉 饶云波

副主编 凌 艳

策划编辑 蔡 喆 李明富

责任编辑 蔡 喆

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhpresse@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787 mm×960 mm 1/16 印张:12.75 字数:286 千字

2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-800-5 定价:20.00 元



高职高专“十一五”示范教材

(美术类学习指导)

编委会

主任

张应辉

副主任

(按姓氏笔画排序)

王海春

汤勇

杨闯

林训超

饶云波

郭杰

策划编委

李明富

《软件工程技术》编委

(按姓氏笔画排序)

王 龙	王 洵
王 津	刘 峰
刘洪涛	刘晋州
李德智	麦兴隆
卓能文	柴 晟
凌 艳	曹 毅
曾 鹏	曾伟一
谭 力	谭 飞

前言

软件已经成为科学和技术各个领域及工业和社会各个部门不可缺少的重要部分。目前软件技术的发展还远远落后于硬件技术,软件开发依然面临着过分依赖人工、软件无法重用、开发大量重复和生产率低下等问题。软件是人类创造出来的逻辑产物,有待于新的理论与实践的探索。软件工程是一门旨在开发满足用户需求、及时交付、不超过预算和无故障的软件的学科,它以质量保证为基础,包括过程、方法和工具三个要素。软件工程涵盖了软件生存周期的所有阶段,包括计划、分析、设计、实现、测试、集成、交付和维护等。本书将帮助读者认识以下问题:

① 什么是软件工程?当前的软件开发过程处于什么水平?

② 怎样做才能改进软件开发过程?

③ 怎样才能保证软件的质量?

④ 软件工程从哪里开始,到哪里结束?

全书共 8 章。

第 1 章是软件工程基础,主要讲述软件工程的定义,软件工程的作用,软件的开发方法,软件的支持过程以及软件的管理过程等。

第 2 章是软件生存周期及开发模型,主要讲述软件生存周期概论,瀑布模型,增量模型,迭代模型,原型模型以及螺旋模型等。

第 3 章是软件项目开发计划,主要讲述项目启动,项目经理及职责,软件项目工作任务分解,软件项目进度计划,软件项目的配置管理,软件项目的质量管理以及开发过程的控制等。

第 4 章是软件需求,主要讲述可需求分析的重要性、任务、目的,需求分析方法及工具,需求管理,需求分析文档以及软件需求阶段的项目管理工作等。

第 5 章是软件设计,主要讲述可设计方法概论,体系结构设计,数据库设计及工具以及软件设计阶段的项目管理工作等。

第 6 章是软件实现,主要讲述软件实现方法,源程序设计风格,屏幕界面设计,帮助信息,处理流程设计,软件实现文档以及软件实现阶段的项目管理工作等。

第 7 章是软件测试,主要讲述软件测试概论,测试用例的设计,软件测试文档以及软件测试阶段的项目管理工作等。

第 8 章是软件维护,主要讲述软件维护基础,软件维护过程,软件维护方法,软件维护文档

以及软件维护阶段的项目管理工作等。

关于软件工程技术的书籍,目前出版的比较多,但要一本适合高职高专院校的读者的书很难。本书列入“高职高专‘十一五’规划示范教材”。全书遵从高职高专教学规律。教材的编写,遵循“实用技术为主、工程实践为线、侧重主流产品”的指导思想,立足于“看得懂、学得会、用得上”的策略。读者只有通过具体的实践,才能加深对所学基础知识的理解,只有经历了实践的全过程,才能系统地掌握各个环节的基本技能。本书是结合作者多年的教学经验而编写出来的,语言通俗易懂,介绍由浅入深,内容丰富翔实,突出“实践性、实用性、创新性”。

本书主编由张应辉、饶云波担任,副主编由凌艳担任,第1,2,8章由凌艳编写,第3,4,7章由张应辉编写,第5,6章由饶云波编写。

本书的出版得到了成都东软信息技术学院的协助,并得到了北京航空航天大学出版社、成都东软信息学院计算机系的大力支持和帮助。周明天教授对本书提出了很好的意见。很多老师、学生和软件工程技术工作人员还通过网站和论坛给予编者很多建议和资源,在此也向这些站点、平台的所有者和参与者表示真诚的感谢。

由于作者水平有限,加之成书仓促,书中如有不妥与错误之处,恳请读者不吝指正,以便在本书再版时修改和充实。

作 者

2006年7月

目 录

第 1 章 软件工程基础	1
1.1 软件工程的定义	1
1.1.1 软件的概念	1
1.1.2 软件危机	4
1.1.3 产生软件危机的原因	5
1.1.4 解决软件危机的途径	6
1.1.5 软件工程的定义	6
1.1.6 软件工程的基本内容和目标	8
1.1.7 软件工程的原则	10
1.2 软件工程的作用	11
1.3 软件的开发方法	11
1.3.1 模块化方法	12
1.3.2 结构化方法	12
1.3.3 面向数据结构的开发方法	14
1.3.4 面向对象的方法	15
1.3.5 可视化开发方法	16
1.4 软件的支持过程	16
1.4.1 文档开发	16
1.4.2 软件配置管理	17
1.4.3 软件质量保证	18
1.5 软件的管理过程	19
1.5.1 软件项目产品的特点	19
1.5.2 软件项目管理的内容	20
1.6 本章小结	21
1.7 习 题	22
第 2 章 软件生存周期及开发模型	23
2.1 软件生存周期概论	23

2.1.1	软件定义	24
2.1.2	软件开发	26
2.1.3	软件使用、维护和退役	28
2.2	瀑布模型	29
2.2.1	瀑布模型的基本原理	29
2.2.2	瀑布模型的缺点	30
2.2.3	改进的瀑布模型	31
2.3	增量模型	32
2.3.1	增量模型概述	32
2.3.2	增量模型分析	33
2.4	迭代模型	34
2.4.1	迭代模型的优点	35
2.4.2	迭代模型的缺点	35
2.4.3	受控的迭代模型	36
2.5	原型模型	38
2.5.1	快速原型模型	39
2.5.2	原型进化模型	40
2.5.3	原型模型的优点	41
2.6	螺旋模型	42
2.6.1	螺旋模型概述	42
2.6.2	螺旋模型分析	43
2.7	本章小结	44
2.8	习题	45
第3章	软件项目开发计划	47
3.1	项目启动	47
3.1.1	项目概述	47
3.1.2	项目启动过程	47
3.1.3	项目章程	48
3.2	项目经理及职责	49
3.2.1	项目经理概述	49
3.2.2	项目经理的责任	49
3.2.3	项目经理的权力	50
3.2.4	项目经理应具备的能力	51

3.3 软件项目工作任务分解	51
3.3.1 工作分解结构	52
3.3.2 WBS 分解的作用	53
3.4 软件项目进度计划	54
3.4.1 进度计划的目的与内容	54
3.4.2 制定进度计划的步骤与技术	54
3.4.3 进度表中的资源分配	56
3.4.4 活 动	56
3.4.5 项目计划的变更方法	57
3.5 软件项目的配置管理	58
3.5.1 软件配置管理的概念	58
3.5.2 配置管理功能概述	58
3.5.3 配置管理活动的作用	59
3.5.4 版本控制	59
3.5.5 变化控制	60
3.6 软件项目的质量管理	61
3.6.1 质量的定义	61
3.6.2 项目质量管理体系	61
3.6.3 软件质量度量	64
3.6.4 软件质量计划编制	64
3.6.5 软件项目质量审核	65
3.7 开发过程的控制	65
3.7.1 项目管理控制的概念	65
3.7.2 项目控制的目标与要求	66
3.7.3 项目控制的基本步骤	66
3.7.4 开发过程的控制工具	66
3.8 本章小结	68
3.9 习 题	68
 第 4 章 软件需求	 70
4.1 需求分析的基本概念	70
4.1.1 需求的定义与分类	70
4.1.2 需求分析的目的与任务	71
4.2 需求管理过程	72

4.2.1	需求获取	72
4.2.2	需求分析	73
4.2.3	规格说明	75
4.2.4	需求验证	77
4.2.5	需求变更	78
4.3	需求分析方法	79
4.3.1	结构化分析法	79
4.3.2	面向对象分析法	86
4.4	需求分析文档	87
4.4.1	需求文档的目的	87
4.4.2	需求文档的特性与风格	87
4.4.3	需求文档的内容及体系结构	88
4.5	本章小结	89
4.6	习 题	89
第 5 章 软件设计		90
5.1	分析与设计的关系	90
5.2	软件设计原则	91
5.2.1	模块化与模块独立性	91
5.2.2	抽象与细化	96
5.2.3	信息隐藏	97
5.3	体系结构设计	97
5.3.1	系统结构图	98
5.3.2	层次图和 HIPO 图	99
5.3.3	面向数据流的设计	100
5.4	过程设计	106
5.4.1	程序流程图	106
5.4.2	盒图(N-S图)	107
5.4.3	判定表	108
5.4.4	判定树	109
5.5	用户界面设计	109
5.5.1	用户界面设计问题	110
5.5.2	用户界面设计过程	111
5.5.3	界面设计指南	111

5.6 设计规格说明书	112
5.7 习 题	113
第 6 章 软件实现	114
6.1 编程语言介绍	114
6.1.1 程序设计语言的基本概念	114
6.1.2 程序设计语言的种类和特点	115
6.2 程序设计风格	118
6.2.1 命名约定	119
6.2.2 变量、常量和数据类型	120
6.2.3 表达式和基本语句	121
6.2.4 一致性和习惯用法	128
6.2.5 注 释	128
6.2.6 空 行	130
6.2.7 对 齐	131
6.3 软件实现阶段的项目管理工作	133
6.4 本章小结	136
6.5 习 题	137
第 7 章 软件测试	138
7.1 软件测试概论	138
7.1.1 软件测试的基本概念	138
7.1.2 常用的软件测试的方法	140
7.2 软件测试用例的设计	144
7.2.1 软件测试用例设计概述	144
7.2.2 白盒测试用例设计方法	146
7.2.3 黑盒测试用例设计方法	151
7.2.4 面向对象软件测试用例的设计	157
7.2.5 测试用例的组织 and 跟踪	157
7.3 软件测试文档	157
7.3.1 测试文档的作用	158
7.3.2 测试文档的类型	158
7.3.3 主要软件测试文档	161
7.4 软件测试阶段的项目管理工作	167

7.5 本章小结	169
7.6 习 题	169
第 8 章 软件维护	171
8.1 软件维护基础	171
8.1.1 软件维护的概念	171
8.1.2 软件可维护性	172
8.1.3 影响维护的因素	173
8.1.4 软件维护的策略	174
8.1.5 软件维护的特点	175
8.1.6 软件维护的技术	176
8.1.7 软件维护的副作用	177
8.2 软件维护过程	178
8.2.1 维护组织	178
8.2.2 维护的报告与审核	179
8.2.3 维护的工作流程	180
8.2.4 维护记录	181
8.2.5 维护评价	182
8.3 软件维护方法	182
8.3.1 改正性维护	182
8.3.2 完善性维护	182
8.3.3 适应性维护	183
8.3.4 预防性维护	183
8.4 软件维护文档	184
8.4.1 维护活动评价	184
8.4.2 维护文档与编写	184
8.5 软件维护阶段的项目管理工作	186
8.5.1 管理层的责任	186
8.5.2 提高维护生产率	187
8.5.3 沟 通	188
8.6 本章小结	189
8.7 习 题	189
参考文献	191

第 1 章 软件工程基础

本章教学目标

- ◇ 理解软件工程的定义及作用
- ◇ 掌握软件的开发方法和支持过程
- ◇ 了解软件的管理过程

软件遍及当今的世界,人们有时把软件在使我们的生活更舒适、有效率的过程中所扮演的重要角色视为当然。事实上,现在软件直接或者间接地为人们生活的各方面服务,包括那些影响人的健康和财富的重要系统。因此,软件工程显得比以往更加重要。好的软件工程的实施必须确保软件在引导人们的生活中做出积极的贡献。本章重点介绍有关软件工程的基础知识,使读者对软件工程有一个初步的认识。

1.1 软件工程的定义

1.1.1 软件的概念

软件是一种产品,同时又是开发和运行产品的载体。作为一种产品,它表达了由计算机硬件体现的计算潜能。不管是驻留在设备中,还是在主机中,软件是一个信息转换器,能够产生、管理、获取、修改、显示或转换信息。这些信息可以很简单,如 1 bit,也可以很复杂,如多媒体信息。作为开发运行产品的载体,软件是计算机工作和信息通信的基础,也是创建和控制其他程序的基础。

信息是 21 世纪最重要的产品,软件充分地体现了这一点。软件处理数据,并使这些数据更为有用。软件管理商业信息增强了商业竞争力,它不仅提供了通往全球信息网络的途径,而且提供了以各种形式获取信息的手段。

1. 软件的含义

随着计算机技术的发展,对软件在不同阶段有不同的认识。计算机发展的初期,硬件的设计和生是主要问题,那时的所谓软件就是程序,甚至是机器指令程序,它们处于从属的地位。软件的生产方式是个体手工方式,设计过程是在人的头脑中完成的,程序的质量完全取决于个人的编程技巧。其后,人们认识到在计算机上增加软件的功能会使计算机系统的功能大大提高,因此在研制计算机体系时既要考虑其硬件又要考虑与之配套的软件,方可开始编制一些大型程序系统。这时的生产方式为互助合作的手工方式,软件基本上就是程序加说明书。



随后社会对计算机提出了更高的要求,有的大型系统的设计和生的工作量高达几千人/年(1个人工作1年其工作量为1人/年),指令数百万条。现在,软件在计算机系统中的重要越来越大,而且这种趋势还在增加。人们感到传统的软件生产方式已经不再适应发展的需要,因此提出把工程学的基本原理和方法引进到软件生产中,把软件生产分成几个阶段,每个阶段都有严格管理和质量检验,研制软件设计和生产的方法及工具,并用书面文件作为共同遵循的依据,这时软件的含义就成了“文档+程序”。

现在对软件的正确理解应该是,软件是计算机系统中与硬件相互依存的部分,包括程序及其相关文档。程序是计算机任务的处理对象和处理规则的描述;文档是为了理解程序所需的阐述性资料。

2. 软件的特点

和硬件相比,软件主要有以下特点。

(1) 表现形式不同

硬件是有形有色、看得见摸得着的,而软件是无形无色、看不见摸不着的。软件正确与否,是好是坏,一直要到程序在机器上运行才能知道,这给设计、生产和管理带来许多困难,生产成本也相当高。

(2) 生产方式不同

软件是人的智力的高度发挥,不同于传统意义上的硬件制造,它没有明显的制造过程,对软件的质量控制必须立足于软件开发过程。

(3) 维护不同

硬件是有损耗的,会产生磨损和老化,使故障率增加甚至自身损坏,解决的办法是换上一个相同的备件。而软件不存在磨损和老化问题,但却存在需要更新的问题。因为在软件的生存期中,一直处于改变(维护)状态,随着针对某些缺陷的维护,可能带来一些新的缺陷,使软件的故障率增加。而软件不像硬件一样有备件,它的维护要比硬件复杂得多。

3. 软件的分类

(1) 按软件的功能

- ◆ 系统软件:能与计算机硬件紧密配合,使计算机系统的各个部件、相关的软件和数据协调高效地工作的软件。如操作系统、数据库管理系统、设备驱动程序以及通信处理程序等。
- ◆ 支撑软件:是协助用户开发应用软件的工具性软件,包括帮助程序人员开发软件产品的工具,也包括帮助管理人员控制开发进程的工具。
- ◆ 应用软件:在特定领域内开发,为特定目的服务的软件。

(2) 按软件的工作方式

- ◆ 实时处理软件:在时间或数据产生时对其立即处理,并及时反馈信号以控制需要检测的过程的软件。实时处理软件主要包括数据采集、分析、输出三个部分。



- ◆ 分时软件:允许多个联机用户同时使用计算机的软件。
- ◆ 交互式软件:实现人-机通信的软件。
- ◆ 批处理软件:把一组输入作业或一批数据以成批处理的方式一次运行,按顺序逐个处理的软件。

(3) 按软件服务对象的范围

- ◆ 项目软件:也称定制软件,是为某个特定客户或少数客户开发的软件,如军用防空指挥系统、卫星控制系统等。
- ◆ 产品软件:直接提供给市场或是为众多用户服务的软件,如文字处理软件、财务处理软件、人事管理软件等。

(4) 基于软件规模的划分

根据开发软件所需的人力、时间以及完成的源程序行数,可划分为下述6种不同规模的软件。

- ◆ 微型软件:1个人在几天之内完成的、程序不超过500行语句且仅供个人专用的软件。通常这类软件没有必要作严格的分析,也不必有完整的设计和测试资料。
- ◆ 小型软件:1个人半年之内完成的2000行以内的程序。这种程序通常没有与其他程序的接口,但需要按一定的标准化技术、正规的资料书写以及定期的系统审查,只是没有大题目那样严格。
- ◆ 中型软件:5个人以内在1年多时间里完成的5000~50000行的程序。中型软件的实施过程中开始出现了软件人员之间、软件人员与用户之间的联系及协调的配合关系问题。因而计划、资料书写以及技术审查需要比较严格地进行。在开发中使用系统的软件工程方法是完全必要的,这对提高软件产品质量和程序人员的工作效率起着重要的作用。
- ◆ 大型软件:5~10人在两年多的时间里完成的5~10万行的程序。参加工作的软件人员需要按二级管理。在任务完成过程中,人员调整往往不可避免,因此会出现对新手的培训和逐步熟悉工作的问题。对于这样规模的软件,采用统一的标准,实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性,往往在开发的过程中出现一些事先难以做出估计的不测事件。
- ◆ 甚大型软件:100~1000人参加,用4~5年时间完成的具有100万行程序的软件项目。这种甚大型项目可能会划分成若干个子项目,每一个子项目都是一个大型软件,各子项目之间具有复杂的接口。例如,实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统通常有这样的规模。很显然,如果这类问题没有软件工程方法的支持,它的开发工作是难以想像的。
- ◆ 极大型软件:2000~5000人参加,10年内完成的1000万行以内的程序。这类软件很少见,往往是军事指挥、弹道导弹防御系统等。



可以看出,规模大、时间长、很多人参加的软件项目,其开发工作必须要有软件工程的知识指导。而规模小、时间短、参加人员少的软件项目也应有软件工程概念,遵循一定的开发规范,其基本原则是一样的。

(5) 基于软件失效的影响进行划分

工作在不同领域的软件,为适应其不同的需求,在运行中对可靠性也有不同的要求。如有的软件在工作中出现了故障,造成软件失效,但可能给软件整个系统带来的影响不大,虽然可能带来一些不便,却能勉强工作。但有的软件一旦失效,可能酿成灾难性后果,其严重损失难以挽回。如控制载人飞行物的软件,如果不能正常工作,可能以人的生命为代价。

事实上,随着计算机进入国民经济等各个重要领域,其软件的可靠性越来越显得重要。人们一般称这类软件为关键软件,其特点在于:

- ◆ 可靠性质量要求高;
- ◆ 常与完成重要功能的大系统的处理部件相连;
- ◆ 含有的程序可能对人员、公众、设备或设施的安全造成影响,还可能影响到环境的质量和关系到国家的安全和机密。

1.1.2 软件危机

在软件技术发展的阶段,随着计算机硬件技术的进步,计算机的应用越来越广泛,软件的数量急剧膨胀,软件需求日趋复杂。然而软件开发技术的进步却严重滞后于形势发展提出的要求,造成供求关系失调,形成了日益尖锐的矛盾。主要问题有下面几个。

- ◆ 软件的开发费用和进度难以控制。由于缺乏软件开发的经验和有关软件开发数据的积累,使得开发工作的计划很难制定,致使费用常常突破预算,进度计划无法遵循。
- ◆ 开发出来的软件不能满足用户的要求。由于软件需求在开发的初期阶段提得不够明确,或者表达得不确切,理解存在偏差,在开发过程中,软件人员和用户又未能及时交换意见,造成开发后期矛盾的集中暴露。
- ◆ 软件的可维护性差。由于开发过程没有统一的、公认的方法论和规范指导,参加的人员各行其是,设计和实现过程的资料很不完整,或忽视了每个人与其他人的接口,使得软件维护在源代码级上进行,增加了软件维护的难度。
- ◆ 软件质量差。由于未能在测试阶段充分做好检测工作,致使提交用户的软件隐藏大量错误,软件运行的可靠性差。

这些问题导致一些软件开发项目耗费了大量的人力、物力、财力,而失败的软件开发项目却屡见不鲜。例如,美国公司开发的 IBM 360 机的操作系统,耗资几千万美元,动用 5 000 多人,花费了几年才完成,但交付使用后仍不断发现错误。

失败的教训引起了人们对这些问题的重视,1968 年北大西洋公约组织的计算机科学家在联邦德国召开的国际学术会议上第一次提出了“软件危机(software crisis)”一词。

