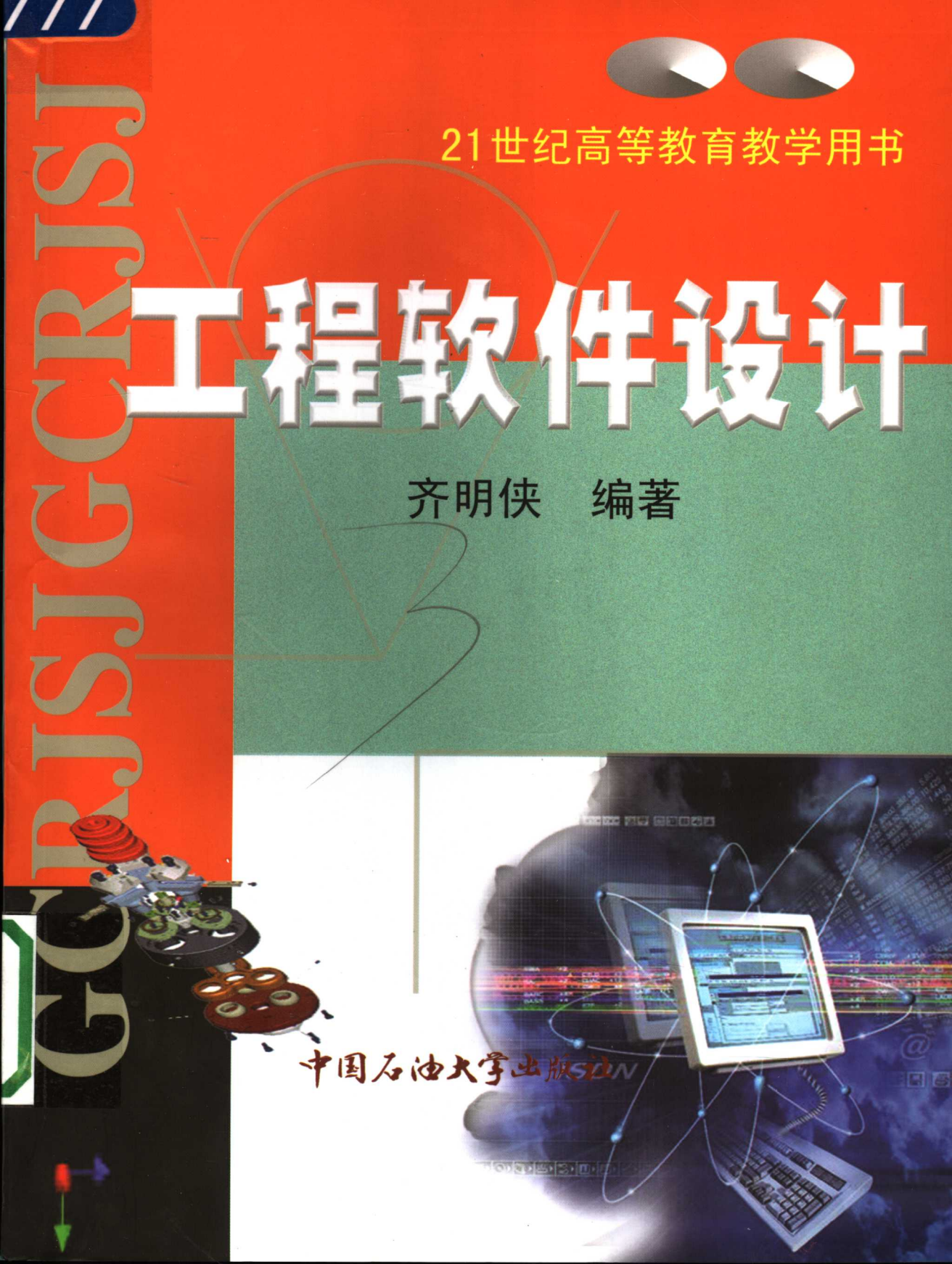


21世纪高等教育教学用书

工程软件设计

齐明侠 编著

中国石油大学出版社



21世纪高等教育教学用书

工程软件设计

齐明侠 编著

中国石油大学出版社

内 容 提 要

本书系中国石油大学“十五”规划教材,适合于高等院校机械工程专业学生使用,也可作为机械工程技术人员的参考用书。本书最大的特点是注重原理和应用相结合,着重培养学生进行工程实用软件开发的能力,为此加入了部分实用的程序。

本书的主要内容包括:软件工程概述、软件工程基础、计算机程序语言简介、数据接口、图形显示、人机界面、程序模型建立等,并附上了 TC2.0 语言编程简介、四连杆机构的工程软件设计实例和 ASCII 码表。

图书在版编目(CIP)数据

工程软件设计/齐明侠编著. —东营:中国石油大学出版社,2006.2

ISBN 7-5636-2178-4

I. 工... II. 齐... III. 软件设计—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2006)第 009529 号

书 名: 工程软件设计

作 者: 齐明侠

责任编辑: 阙青兵

封面设计: 傅荣治

出 版 者: 中国石油大学出版社(山东 东营 邮编 257061)

网 址: <http://cbs.hdpu.edu.cn>

电子信箱: bianwn@hdpu.edu.cn

排 版 者: 中国石油大学出版社排版中心

印 刷 者: 青岛星球印刷有限公司

发 行 者: 中国石油大学出版社(电话 0546-8392563)

开 本: 180×235 印张: 8.75 字数: 174 千字

版 次: 2006 年 2 月第 1 版第 1 次印刷

定 价: 13.80 元

前言

Foreword Foreword Foreword Foreword

计算机的飞速发展,使得计算机进入了社会的各个行业,进入了千家万户,计算机已经成为工程技术人员的得力助手,发挥着越来越重要的作用。为帮助机械工程专业学生更好地掌握计算机的应用,提高他们编制实用软件的水平,我校为机械工程专业学生开设了工程软件设计这门课程。为满足教学需要,编写了此书。

全书共分七章,第一章为概述;第二章介绍了软件工程的基础,以帮助非软件专业人员了解工程软件的基本原理和方法,用于指导工程软件的编制;第三章介绍了各种程序语言的特点和使用;第四章介绍了计算数据的各种表示方法及计算机的输入、输出接口的使用,包括键盘、针式打印机、鼠标的使用和汉字的显示;第五章介绍了计算机显示系统的基本原理和操作,介绍了 BMP 位图文件;第六章介绍了图形用户界面;第七章为建模的基本知识。附录中介绍了 TC 的使用和四连杆机构计算等内容。

本书在编写过程中力求尽可能地满足机械工程技术人员的需要。有关软件工程的著作很多,一般都为计算机类或软件类的专著,但考虑到机械工程技术人员的往往不是专业的软件编制者,而在工作中经常要使用计算机进行计算、控制、作图及输入/输出等,这时往往需编制一些小型软件,为满足急需,本书的编写目的就是加强对机械工程专业学生进行工程实用软件开发的指导,帮助学生更好地使用计算机为工程服务。本书中除了介绍工程软件的基本原理外,其余内容都是针对编制工程软件时的一些常用需求来编写的,并且给出了一定数量的例题。同时编者还用 C 语言编写了部分短小的实用程序(可来函索取),一方面可加深对书中内容的理解,另一方面也希望起到抛砖引玉、举一反三的目的。

本书由齐明侠编写。在编写过程中,沈蓉对文稿的校对做了许多工作,教务处和出版社给予了大力的支持和帮助,在此表示衷心的感谢!

由于编者的水平和经验有限,错误在所难免,敬请同行及读者提出宝贵意见!

编者

2005-12-07

目 录

Catalog Catalog Catalog Catalog

第一章 概述	(1)
§ 1-1 软件与软件危机	(1)
§ 1-2 软件工程	(4)
§ 1-3 工程软件及其设计	(7)
第二章 软件工程基础	(9)
§ 2-1 软件生存期	(9)
§ 2-2 可行性研究	(14)
§ 2-3 需求分析	(17)
§ 2-4 总体设计	(19)
§ 2-5 详细设计	(25)
§ 2-6 编码	(30)
§ 2-7 测试	(33)
§ 2-8 维护	(39)
第三章 计算机程序语言简介	(42)
§ 3-1 程序设计语言基础	(42)
§ 3-2 通用语言	(43)
§ 3-3 数据库语言	(50)
§ 3-4 人工智能语言	(51)
§ 3-5 其他语言	(54)
第四章 数据接口	(56)
§ 4-1 数据的表示与存储	(56)
§ 4-2 键盘输入接口	(58)
§ 4-3 汉字处理	(60)
§ 4-4 数据输出	(64)
§ 4-5 鼠标	(69)
第五章 图形表示	(73)
§ 5-1 计算机图形显示原理	(73)
§ 5-2 VGA 图形操作技术	(83)
§ 5-3 曲线的表示与绘制	(90)
§ 5-4 图形文件	(94)

第六章 人机界面	(97)
§ 6-1 人机界面的要求	(97)
§ 6-2 图形用户界面	(99)
§ 6-3 菜单设计	(100)
第七章 程序模型建立	(104)
§ 7-1 数理方程模型	(104)
§ 7-2 数理统计模型	(108)
§ 7-3 最优化模型	(109)
§ 7-4 人工智能模型	(111)
§ 7-5 数据库模型	(113)
§ 7-6 数值计算模型	(116)
§ 7-7 迭代模型	(117)
附录 1 TC2.0 语言编程简介	(122)
附录 2 工程软件设计——四连杆机构	(125)
附录 3 ASCII 码表	(132)
参考文献	(133)

第一章 概 述

§ 1-1 软件与软件危机

一、软件

1. 软件的发展

世界上第一台计算机 ENIAC (Electronic Numerical Integrator And Calculator) 于 1946 年 2 月 14 日在美国宾夕法尼亚大学诞生, 运算速度为加法 5 000 次/秒、乘法 56 次/秒, 比继电器计算机快一千倍, 比手工计算快 20 万倍。当时给它出的第一道题是核子物理方面的, 算了两个小时, 而一个优秀的计算员至少要算 100 年以上。

我国从 1956 年开始研究计算机, 1958 年研制出第一台电子管计算机, 1964 年研制成功晶体管计算机, 1971 年研制成功集成电路计算机, 1983 年研制成功运算速度为 1 亿次/秒的“银河 I”巨型机, 1997 年研制成功运算速度为 130 亿次/秒的“银河 III”巨型机。

自第一台计算机出现之后就有了程序的概念, 程序是软件的前身。软件的发展经历了三个阶段: 第一阶段在 20 世纪 50 ~ 60 年代, 为程序设计阶段, 使用的语言为机器语言或汇编语言, 软件为小型规模, 一般由个人开发; 第二阶段在 60 ~ 70 年代, 为程序系统阶段, 使用的语言为高级语言, 软件为中小型规模, 由开发小组开发; 第三个阶段在 70 年代之后, 为软件工程阶段, 由开发小组或开发机构完成软件的研制工作。

2. 软件定义

软件 (Software) 是计算机系统中与硬件 (Hardware) 相互依存的另一部分, 它是包括程序、数据及其相关文档的完整组合。按照 IEEE 组织 1983 年的定义: “软件为计算机程序、方法、规则、相关文档资料以及在计算机上运行所必须的数据”, 所以一个软件应包括三个组成部分——程序、数据和文档。其中程序是按事先设计的功能和性能要求执行的指令序列; 数据是使程序能正常操纵信息的数据结构 (数字); 文档是与程序开发、维护和使用有关的图文材料。我国国家标准《GB/T 11457—89 软件工程术语》中定义“软件是指与计算机操作有关的计算机程序、规程、规则以及任何与之有关的文件”。

3. 软件分类

由于计算机具有各种功能, 所以也就有各式各样的软件, 因此软件的分类也就各式各样。

按软件的功能划分可将软件分为系统软件、支撑软件与应用软件。系统软件是指能与计算机硬件紧密结合在一起,使计算机系统各个部件、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,系统软件是计算机系统不可缺少的一个组成部分;支撑软件是协助用户开发软件的工具性软件,包括帮助程序人员开发软件产品的工具,也包括管理人员控制开发的进程工具,如文本编辑程序、图形软件包、编译程序等;应用软件是在特定领域开发,为特定目的的服务的一类软件。应用软件的种类繁多,能极大地提高生产率和产生巨大的经济效益,如商业数据处理软件、工程与科学计算中的数值计算软件、机械设计制造中的CAD/CAM软件、智能产品的嵌入软件、人工智能软件以及事务管理、办公自动化、计算机辅助教学(CAI)、游戏软件等。

按软件的规模可将软件分为微型、小型、中型、大型、甚大型、极大型等,见表1-1。

表 1-1 软件规模

类 别	参加人数	研制期限	产品规模 (源程序行数)	应用举例
微 型	1	1~4 周	0.5k	小题目
小 型	1	1~6 月	1k~2k	数值计算、数据处理
中 型	2~5	1~2 年	5k~50k	应用程序、系统程序
大 型	5~20	2~3 年	50k~100k	编译程序、分时系统实时控制、应用软件包
甚大型	100~1 000	4~5 年	1M	实时处理、远程通讯操作系统、数据库管理
极大型	2 000~5 000	5~10 年	1M~10M	军事指挥、弹道防御

按软件的工作方式可将软件分为实时处理软件、分时软件、交互式软件和批处理软件等。

按软件服务对象的范围可将软件分为项目(接受委托)软件和产品(供应市场)软件。

按软件的使用频度可将软件分为经常使用软件与不经常使用(甚至一次使用)软件。

按软件失效的影响可将软件分为一般软件和关键(如金融、交通、航天、军事等)软件。

二、软件危机

1. 软件危机的表现

在程序系统阶段就开始出现了软件危机。软件危机是指在计算机软件开发和维护过程中所遇到的一系列严重问题。这些问题不仅仅是“不能正常运行的”软件才具

有的,实际上几乎所有软件都不同程度地存在这些问题。

软件的社会化改变了人们对软件的认识。早期,人们把程序设计看成是一种任人发挥创造才能的技术领域,当时的人们一般认为只要能在计算机上得出正确的结果,程序的写法可以不受约束,而且只有那些通篇充满了技巧及使用了许多窍门(别人很难看懂)的程序才是好的程序。这是因为当时计算机的速度、容量有限,软件质量的目标定为内存省、代码短、速度快。当计算机广泛应用之后,使得软件社会化了。对于稍大一些的程序或在较长一段时间为许多人所使用的程序,人们要求这些程序容易看懂,容易使用并容易修改和扩充,程序从“艺术品”变成了为广大用户接受的工程化产品,这时程序中难以理解的技巧就成了有害的东西。

随着计算机硬件技术的进步,计算机的容量、速度和可靠性有了明显的提高,生产成本降低了,于是对软件提出了发展的要求,要求开发复杂的大型软件。硬件为机械化自动化生产,而软件为手工作坊式生产,二者之间产生了比较尖锐的矛盾。软件适应不了硬件的发展,反映在如何开发软件(对软件的需求不断增长)和如何维护软件(要维护的软件数量不断膨胀)这两个方面。主要表现有:

(1)软件开发无计划性。由于缺乏开发的经验和有关开发数据的积累,使得开发计划难以确定,时间经费都不能很好的预算,用户对开发者无法满意。

(2)软件的需(要)求不充分。用户的需求是软件开发的依据,但在开发初期往往提得不够明确或是未能得到确切的表达,使得一些问题到开发后期才得以暴露,解决问题变得非常困难。

(3)软件的开发过程无规范。软件的开发过程中缺少充分的文字说明,给软件的维护造成困难,有些软件常常是不可维护的。

(4)软件产品无测评手段。这将造成软件质量无法保证,软件的运行也无法看清楚。

(5)软件成本在计算机系统的总成本中所占的比例逐年上升。据统计软件的成本已占计算机系统开发成本的90%以上。

(6)软件开发速度的提高跟不上计算机的发展和普及速度。

2. 解决软件危机的途径

因为软件危机是由于软件生产的特殊条件造成的,所以要解决软件危机,就必须引进先进的生产管理规范,提高软件的生产质量。如:

(1)有组织地进行软件生产;

(2)总结经验教训,研究探索有效开发软件的技术和方法;

(3)开发、使用更好的软件工具。

总之为了解决软件危机,既要有技术措施,又要有必要的组织管理措施,于是产生了软件工程学科。

§ 1 - 2 软件工程

一、软件工程的定义

软件工程是指导计算机软件开发和维护的工程学科,它采用工程的概念、原理、技术和方法来开发和维护软件,把经过时间验证是正确的管理技术与当前能够得到的最好的技术和方法结合起来。“软件工程”于 1968 年在联邦德国召开的国际会议上正式被提出并使用。按照 IEEE 1983 年的定义:“软件工程是开发、运行、维护和修复软件的系统方法。”

二、软件工程的基本原理

著名的软件工程专家 B. W. Boehm 根据其在软件工程领域的经验,于 1983 年提出了 7 条软件工程的基本原理,为大家所接受,简单介绍如下。

1. 用分阶段的生命周期计划严格管理

有人统计发现,不成功的软件项目有一半左右是由于计划不周造成的。

在软件开发与维修的漫长的生命周期中,需要完成许多性质各异的工作。这条基本原理意味着应该把软件的生命周期划分成若干个阶段,并相应地制定切实可行的计划,然后按照计划对软件的开发与维护工作进行管理。

如在软件整个生命周期中应该制定并严格执行各类计划,包括项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划、运行维护计划。

2. 坚持进行阶段评审

软件的质量保证工作不能等到编码阶段结束之后再进行,因为:① 据统计,软件的错误中设计错误占 63%,编码错误占 37%,大部分错误是在编码之前造成的;② 错误发现和改正得越晚,所付出的代价也越高,且成非线性变化(图 1-1)。因此,必须在每个阶段进行严格的评审,以便尽早发现在软件开发过程中所犯的错误。

3. 实行严格的产品控制

在软件开发过程中不应随意改变需求,但是随着外部环境的变化,改变需求往往又是难免的,必须用科学的产品控制技术来顺应这种要求,按一定的规程进行产品修改。

4. 采用现代程序设计技术

采用先进的程序设计技术既可以提高软件的开发和维护效率,又可以提高软件的质量。

5. 结果应清楚地审查

软件是看不见的逻辑产品,可见性差,难以准确度量,从而使软件开发过程比一般

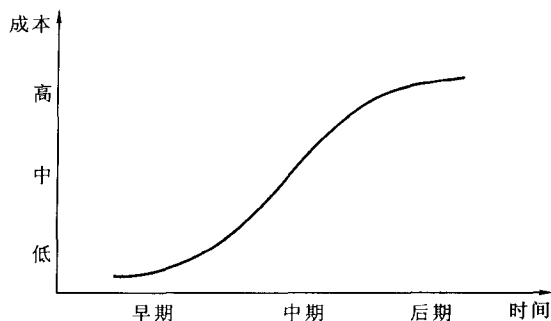


图 1-1 软件改动与成本间的关系

产品的开发过程更难于评价和管理。因此必须制定开发目标,规定产品标准,以利于产品的审查。

6. 开发小组的人员应该少而精

这条基本原理的含义是软件开发小组的组成人员的素质应该高,而人数不宜过多。人数多会导致通讯接口和通讯开销的增加,如开发小组人数为 N ,则可能的通讯路径为 $N(N-1)/2$ 。

7. 承认不断改进软件工程实践的必要性

这条基本原理的含义是指人们不仅要积极主动地采用新的软件技术,而且要注意不断地总结经验。

三、软件工程的基本目标

软件工程的基本目标是开发出用户满意的高质量软件,具体指标有:

- (1) 低成本——付出较低的开发成本;
- (2) 高可靠性——达到要求的软件功能;
- (3) 高性能——取得较好的软件性能;
- (4) 易移植——开发的软件易于移植;
- (5) 易维护——需要较低的维护费用;
- (6) 按时交付——能按时完成开发工作。

四、软件生存期

一般来说,软件的生命周期由软件的定义、软件的开发和软件的维护三个时期组成,每个时期再划分为若干个阶段。

软件定义时期的任务是确定软件开发工程必须完成的总目标,确定工程的可行性,导出实现工程目标应该采取的策略及系统必须完成的功能,估计完成该项工程需要的资源和成本。这个时期的工作通常又称为系统分析,由系统分析员负责完成。软件定义时期包括问题定义、可行性研究和需求分析三个阶段。

软件开发时期的主要任务是具体设计和实现在前一个时期定义的软件,一般由总体设计、详细设计、编码和单元测试、综合测试四个阶段组成。前两个阶段称为系统设计,后两个阶段称为系统实现。

软件维护时期的主要任务是使软件持久地满足用户的需要,包括对软件错误的改正、当环境变化时的修改、当有新要求时的改进。一般不再对维护时期划分阶段,但每次维护活动实质上都是一次压缩和简化了的软件定义和开发过程。

五、技术审查和管理复审

软件中的一个非常特殊的现象是必须承认软件中有存在错误的可能,大型软件中存在错误几乎是不可避免的,因此对软件进行技术审查是非常必要的。

一项大型软件开发工程需要花费大量的人力、物力和资金,而且投入运行后需要的维护费用通常大大超过开发费用,因此必须慎重对待。

软件工程的一个基本方法就是在每个阶段结束前都进行正式的技术审查和管理复审。

技术审查是保证软件质量的重要措施。由于人的认识不可能百分之百地符合实际,因此在软件生命周期每个阶段的工作都可能发生错误,而且前一阶段的成果是后一阶段工作的基础,前一阶段的错误自然会导致后一阶段的工作结果中有相应的错误,因此,错误就会积累起来。此外,后一阶段的工作一般是把前一阶段的结果进一步具体化,因此,前一阶段中的一个错误可能造成在后一阶段中出现了几个错误,也就是说,软件错误不仅有积累效应,还有“放大”效应。

技术审查也是降低软件成本的一个重要措施。因为在后期改正一个错误比在早期改正同一个错误需要付出的代价高出二至三个数量级,所以越是在早期发现的错误越容易改正,代价越低。

通常成立一个由技术专家组成的审查小组来承担技术审查工作。审查小组的基本成员有四个人——组长、作者和两名评审员。完整的技术审查过程一般分为六个步骤:准备、简要介绍情况、阅读被审查的文档、开审查会、返工、复查。开审查会的目的是发现错误而不是改正错误。

返工后审查小组的组长和作者再聚会就对返工结果进行复查。如果组长表示满意,则审查过程结束;如不满意,组长可要求作者重新返工并且再次复查,必要时组长可以要求重新审查。

通常在技术审查合格之后再进行管理复审,这样,管理人员可以专心地从管理角度对开发工作进行审查,而不必顾及技术问题。管理复审的任务是在软件生命周期的每个重要的里程碑(一般是每个阶段结束时)对工程项目的成本、实际花费的经费、投资回收的前景、项目的进度等经济因素从管理角度进行审查。

§ 1-3 工程软件及其设计

一、工程软件

从广义上讲,任何软件都服务于工程,都是工程软件。狭义上的工程软件是指服务于机械、机电工程的软件,一般为中小型软件。工程软件可分为通用型和专用型两类。通用型软件如机械制图软件 CAD,机械辅助设计软件 CAE、Pro - E,结构分析软件 ANSYS、SAP,系统仿真软件 ADAMS。工程语言 MATLAB 等也可归入通用软件类中,大多数工程软件都属于通用型软件。

工程软件按工作性质可分为以下几类:

- (1) 设计、计算软件——如 CAD、CAM、数值分析、工程计算软件等;
- (2) 监测软件——如数据采集、故障监测软件等;
- (3) 控制软件——如设备控制、过程控制、数控机床软件等;
- (4) 辅助软件——用于辅助处理,如商用软件的前后处理程序等。

工程软件属于软件范畴,具有软件的一切特点和设计要求,除此之外还有自己的特点,如性能专一、目的性强、和硬件的结合密切、注重成本因素等。

二、工程软件的组成及特点

工程软件主要由输入、输出、处理、控制等几个部分组成。

工程软件重视计算机的接口,包括硬件接口和软件接口。硬件接口如计算机的串口、并口、USB 口,计算机内的 A/D、D/A 接口板等;软件接口如各种商用软件的输入/输出接口等等。掌握这些接口既可以使计算机对外进行控制,也可以为商用软件编制专用前、后处理程序,还可使用商用软件的强大功能。

工程软件经常需要直接驱动机械设备或进行过程控制,因此对可靠性有非常高的要求。

工程软件注重数据的输入和输出,特别是数据的可视化,如报表、曲线或者动画等。

工程软件的实用性强。工程软件一切从实际出发,满足实用要求。有时无理论解,只有近似解;有时无最优解,只能得到可行解或满意解。特别是对于优化、迭代、人工神经网络等问题往往受初始值的影响不一定能找到完美的解决方案时,满意解往往为工程所接受。

工程软件的设计实际上就是用软件设计的方法对工程用软件进行设计。特别是对于非软件专业的工程技术人员来讲,一般不需要编制大型及以上软件,因此,本门课

的主要任务是掌握中小型工程用软件的编制原则和方法。本门课的主要内容包括软件工程的基本原理,工程问题的计算模型,计算机的输入、输出(键盘、显示器、打印机)设备的特性和驱动方法及人机交互等内容。有关接口方面的内容将在测控技术及机电系统设计中介绍。

第二章 软件工程基础

§ 2-1 软件生存期

一、软件的生命周期

软件的生命周期由软件的定义、软件的开发和软件的维护三个时期组成。三个时期又可划分为若干个阶段。软件的定义包括问题定义、可行性研究、需求分析三个阶段;软件开发包括总体设计、详细设计、编码和单元测试、综合测试四个阶段;软件的维护一般不再划分阶段。

下面简要介绍一下各个阶段的基本任务和结束标准。

1. 问题定义

问题定义阶段必须要回答的问题是:“要解决的问题是什么?”如果不知道问题是什么就想去解决,显然是盲目的。此步骤经常被忽略,一定要注意。

问题定义阶段是软件生命周期中最短的阶段,一般需要一天甚至更少的时间。在该阶段,通过程序分析员和用户之间的讨论,由分析员完成对问题的理解文档报告。

2. 可行性研究

这个阶段要回答的问题是:“对于上一阶段所确定的问题有行得通的办法了吗?”为了回答这个问题,系统分析员需要进行一次大大压缩和简化了的系统分析和设计过程,也就是较抽象的高层次的分析和设计的过程。可行性研究应该比较简短,这个阶段的任务不是具体解决问题,而是研究问题的范围,探索这个问题是否值得去解,是否有可行的解决办法。

在问题定义阶段提出的对工程目标和规模的报告比较含糊,可行性研究阶段应该导出系统的高层逻辑模型(通常用数据流图表示),并且在此基础上确定更准确、更具体的工程规模和目标,然后分析员要更准确地估计成本和效益。对建议的系统进行仔细的成本和效益分析是这个阶段的主要任务之一。可行性研究的结果是使部门负责人做出是否继续这项工程的决定的重要依据。

3. 需求分析

这个阶段的任务不是具体地解决问题,而是准确地确定“为了解决这个问题,目标系统必须做什么”,主要是确定目标系统必须具备哪些功能。

用户了解他们所面临的问题,知道必须做什么,但是通常不能够完整而准确地表达出他们的要求,更不知道怎样利用计算机解决他们的问题。软件开发人员知道怎样

利用软件实现人们的要求,但是对于特定用户的具体要求并不完全清楚。因此,系统分析员在需求分析阶段必须和用户密切配合,充分交流信息,以得出经过用户确认的系统逻辑模型。通常用数据流图、数据字典和简要的算法表示系统的逻辑模型。

在需求分析阶段确定的系统模型是以后设计和实现目标系统的基础,因此必须准确而完整地体现用户的要求。系统分析员通常都是计算机软件专家,技术人员一般都喜欢很快着手具体设计,然而,一旦分析员开始谈论程序的细节,就会脱离用户,使他们不能继续提出他们的要求和建议。软件工程使用的结构分析的方法为每个阶段都规定了特定的结束标准,需求分析阶段必须提出完整的系统逻辑模型,经过用户确认之后才能进入下一个阶段,这就可以有效地防止和克服急于着手进行具体设计的倾向。

4. 总体设计

这个阶段必须回答的关键问题是:“概括地说,应该如何解决这个问题?”例如,目标系统的一些主要功能是用计算机自动完成还是人工完成;如果是使用计算机,那么是使用批处理方式还是人-机交互方式;信息存储是使用传统的文件系统,还是数据库等等。一般应考虑下面几种可能的方案:

(1)低成本解决方案。系统只能完成最必要的工作。

(2)中等成本解决方案。系统除完成任务外,还具有某些用户无要求但程序员认为是很有价值的附加功能。

(3)高成本的“十全十美”的系统。这样的系统具有用户可能希望的所有功能。

系统分析员应该使用系统流程图或其他工具描述每种可能的系统,估计每种系统的成本和效益,并推荐一个较好的系统。

总体设计的第二项主要任务是设计软件的结构,也就是确定程序的组成模块以及各模块之间的关系。通常用层次图或结构图描述软件结构。

5. 详细设计

总体设计阶段以比较抽象概括的方式提出了解决问题的办法,详细设计阶段的任务是把解法具体化,也就是回答这个关键问题:“应该怎样具体地实现这个系统?”

这个阶段的任务还不是编写程序,而是设计出程序的详细规格说明。这种规格说明的作用类似于其他工程领域中工程师经常使用的工程蓝图,它们应该包括必要的细节,程序员可以根据它们编写出实际的程序代码。

通常用 HIPO(层次图加输入/处理/输出)图或 PDL 语言(过程设计语言)描述详细设计的结果。

6. 编码和单元测试

这个阶段的任务是写出正确的容易理解和维护的程序模块。

程序员应该根据目标系统的性质和实际环境选取一种适当的高级程序设计语言(必要时用汇编语言),把详细设计的结果翻译成用选定的语言书写的程序,并且仔细

测试编写出的每一个模块。

7. 综合测试

这个阶段的关键任务是通过各种类型的测试及相应的调试,使软件达到预定的要求。

最基本的测试有集成测试和验收测试。集成测试是根据设计的软件结构,把经过单元测试检验的模块按某种选定的策略装配起来,在装配过程中对程序进行必要的测试。而验收测试则是按照规格说明书的规定(通常在需求分析阶段确定),由用户(或在用户积极参与下)对目标系统进行验收。

必要时还可以再通过现场测试或平等运行等方法对目标系统进一步测试检验。

为了使用户能够参加各级验收测试,并且在系统投入生产性运行以后能够正确有效地使用这个系统,通常需要以正式或非正式的方式对用户进行培训。

通过对软件的测试结果的分析可以预测软件的可靠性;反之,根据对软件的可靠性要求也可以决定测试和调试过程什么时候可以结束。

应该用正式的文档资料把测试计划、详细测试方案以及测试结果保存下来,作为软件配置的一个组成部分。

8. 软件维护

维护阶段的任务是通过各种必要的维护活动使系统持久地满足用户的需要。通常有四类维护活动:改正性维护,即诊断和改正在使用过程中发现的软件错误;适应性维护,即修改软件以适应环境的变化;完善性维护,即根据用户的要求改进或扩充软件,使之更加完善;预防性维护,即修改软件,为将来的维护活动预先做准备。

虽然没有把维护阶段进一步划分成更小的阶段,但实际上每一项维护活动都应该经过提出维护要求(或报告问题)、分析用户要求、提出维护方案、审批维护方案、确定维护计划、修改软件设计、修改程序、测试程序、复查验收等一系列步骤,因此,实质上是经历了一次压缩和简化了的软件定义和开发的全过程。

每一项维护活动都应该准确地记录下来,作为正式文档资料加以保存。

二、软件生存期模型

1. 瀑布模型

瀑布模型规定了各项软件工程活动,包括问题定义、可行性研究、需求分析、总体设计、详细设计、编码和单元测试、综合测试及运行维护各个阶段,并且规定了它们自上而下、相互衔接的固定次序,如同瀑布流水,逐级下降,如图2-1所示。

每项开发活动均应具有以下特征:

- (1)从上一项活动接受该项活动的工作对象,作为输入;
- (2)利用这一输入实施该项活动应完成的内容;
- (3)给出该项活动的工作成果,作为输出传给下一项活动;