

实用软件工程

郑人杰 编著

清华大学出版社

内 容 简 介

软件工程是近年发展起来的计算机技术新学科。本书是作者在清华大学讲授“软件工程概论”课的讲义基础上编写而成的。从软件的开发、维护和软件管理等方面系统地阐述了软件工程的基本概念和常用的方法。全书以易懂和实用为原则,各章节尽可能结合实例讲述,书后附有软件项目文档的编写大纲。

本书可作为高等学校计算机有关专业的教材或教学参考书,也可供计算机软件人员或计算机用户阅读。

实 用 软 件 工 程

郑人杰 编著

清华大学出版社出版

北京 清华园

清华大学印刷厂印刷

新华书店总店科技发行所发行

开本: 850×1168 1/32 印张: 9.5 字数: 247千字

1991年2月第1版 1991年2月第1次印刷

印数: 00001-10000

ISBN 7-302-00810-8/TP·290

定价: 4.90元

前 言

计算机软件人员水平考试是造就宏大的多层次计算机人才队伍的一项重要措施。从 1990 年起,在全国实施《中国计算机应用软件人员专业技术职务任职资格(水平)考试暂行规定》,资格考试级别分为程序员级(相当于技术员、助理工程师)、高级程序员级(相当于工程师)、系统分析员级(相当于高级工程师)。为配合考试我们在 1987 年和 1990 年曾先后两次组织编写过综合性的考试辅导教材。这些教材适应了广大应考人员的备考急需,同时也为一些科技人员提供了一套系统学习计算机软件知识的资料。但是限于篇幅,在三、四十万字的一本书中包含众多方面的内容,就难免成为一本浓缩了的读物。初学者用它作为自学教材,势必有许多困难。

为满足来自非计算机专业的软件人员系统学习计算机软件知识的迫切要求,我们决定进一步组织力量编写本系列丛书。编写的原则是:

(1) 丛书的选题基本上围绕考试大纲,面向计算机软件人员。每书一个内容,便于选读。

(2) 丛书的编写力争作到深入浅出、简明实用、叙述严谨、适合自学。

(3) 丛书分为程序员、高级程序员及系统分析员三个级别。除去每册各属三个级别之外,也有的书册可供两个相邻的级别使用。

(4) 丛书可分为核心书和参考书两类。核心书反映了考试大纲规定的基本要求,参考书的内容则是在基本要求的基础上的进一步扩展。

为保证丛书的质量,我们聘请的作者均为在计算机软件领域的教学、科研或开发中具有相当的理论基础和丰富实践经验的专家。同时他们还具有一定的软件人员水平考试的教学辅导的实践经验。

我们相信本系列丛书的出版会受到广大读者的欢迎。同时也欢迎对丛书提出宝贵意见。

在本丛书的编写过程中,曾得到中国计算机学会、中国软件行业协会的指导和帮助、清华大学出版社为本书的迅速出版付出了辛勤劳动,在此一并致谢。

丛书编审委员会

目 录

第一章 软件工程概述.....	1
1.1 软件的特点及分类	1
1.2 软件的发展和软件危机	8
1.3 软件开发工程化和软件生存期.....	12
1.4 软件工程项目的目标.....	16
第二章 软件需求分析	21
2.1 需求分析阶段的任务	21
2.2 软件需求分析的原则	26
2.3 分析员和用户的责任	29
2.4 软件规格说明书	30
2.5 结构化分析方法	31
2.6 数据流图	32
2.7 数据词典	40
2.8 数据流图和数据词典应用实例	42
2.9 判定表和判定树	45
2.10 面向数据结构的分析方法	49
2.11 结构化数据系统开发	50
2.12 Jackson 系统开发	57
2.13 结构化分析与设计方法	63
第三章 软件设计	67
3.1 软件设计阶段的任务.....	67
3.2 程序结构与结构图.....	70
3.3 程序内部的联系.....	81

3.4	结构化设计方法.....	91
第四章	详细设计的表达.....	101
4.1	程序流程图	102
4.2	N-S 图	107
4.3	PAD 图	111
4.4	PDL 语言	113
4.5	HIPO 图	122
4.6	选用详细设计表达工具的原则	128
第五章	结构化程序设计与程序设计风格.....	129
5.1	对源程序的质量要求	129
5.2	结构化程序设计	131
5.3	程序设计风格	140
第六章	软件测试.....	152
6.1	软件测试的基本概念	153
6.2	测试用例的设计	166
6.3	软件测试策略	187
第七章	软件维护.....	202
7.1	什么是软件维护	202
7.2	维护工作存在的问题及其分析	206
7.3	可维护性及其度量	209
7.4	软件维护的管理	216
第八章	软件管理.....	220
8.1	软件项目的特点和软件管理的职能	221
8.2	制定计划	227
8.3	建立组织	236
8.4	配备人员	241
8.5	指导与检验	244
8.6	软件配置管理	247

8.7 软件成本估算	255
第九章 软件工程标准和软件产品文档编制.....	264
9.1 什么是软件工程标准化	265
9.2 软件工程标准化的意义	268
9.3 软件工程标准的制定与推行	268
9.4 软件工程标准的层次	270
9.5 我国的软件工程标准化工作	272
9.6 文档的作用和分类	273
9.7 文档编制的质量要求	278
9.8 文档的管理和维护	279
附录 文档编写纲要.....	281
主要参考文献.....	294

第一章 软件工程概述

在近代技术发展的历史上,工程学科的进步一直是产业发展的巨大推动力。传统的工程学科走过的道路已为人们所熟知,水利工程、建筑工程、机械工程、电力工程等对于工农业、商业、交通业的影响是极为明显的。人类在认识和征服大自然的长征中继续前进,近年来人们开始对气象工程、生物工程、计算机工程等有了新的认识。然而,对于工程学这个家族中的另一新成员——软件工程却很不熟悉。其实,这并不是因为它的地位无关紧要,恰恰相反,它对软件产业的形成和发展起着决定性的推动作用。我们说它在计算机系统的发展和应用中至关重要,说它在人类进入信息化社会时成为新兴信息产业的支柱,绝不过分,更非无稽之谈。人们对软件工程不了解,其根本原因是对软件本身认识不清。本章将对软件的特点、软件工程的形成及软件生命期等概念给出简要的介绍,以期使读者从中得到对软件工程最起码的理解。

1.1 软件的特点及分类

“软件”这一名词 60 年代初从国外传来,当时许多人说不清它的确切含意。software 一词确是 soft 和 ware 两字组合而成。有人译为“软制品”,也有人译为“软体”。现在我们都统一称它为软件。对它的一种公认的解释为,软件由三部分组成:程序、数据和文档。即按事先设计的功能和性能要求执行的指令序列,使程序能正常加工信息所需要的数据以及描述程序操作及使用的资料。尽管这个说法并不是计算机软件的精确定义,然而却有助于让我们把它和扩充了含意的广义软件相区别。因为指明某一行业生产技术、管

理制度等的所谓广义软件,今天已经进入我们的社会生活。

为了能全面、正确地理解计算机和软件,有必要让我们来分析一下软件的特点。

(1) 计算机软件是一种逻辑实体,而不是物理实体。因而它具有抽象性。这个特点使它和计算机硬件或是其它工程对象有着明显的差别。我们可以把它记录在纸面上,保存在计算机的存储器内部,也可以保留在磁盘、磁带上,但我们却无法看到软件本身的形态。

(2) 计算机软件在研制、开发活动中被创造出来,但它不能按传统的“生产”含意加以理解。尽管软件开发和硬件制造之间也有着某些相似的步骤。但硬件制造过程中质量因素一直是至关重要的问题,而软件的情况却有很大的不同。软件的研制需要花费很大力气,一旦研制出来,大批生产几乎不花什么成本,是件非常容易做的事,其中的质量因素也比较容易掌握。正是由于这个特点,软件的复制太简单了。于是出现了软件的保护问题。为了使软件研制的复杂劳动受到社会的承认和尊重,必须从技术上和法律上采取有力的措施,对于任意复制软件的行为加以严格的限制。

虽然近年来国内外也都有建立“软件工厂”的说法,但软件工厂毕竟只是为软件开发手段或开发环境创造更加优越的条件,以利于高效地开发软件,并不意味着按硬件生产的模式生产软件。

(3) 软件在长期运行和使用中没有磨损、老化、用旧等问题。任何机械、电子设备在运行和使用中,其失效率大都遵循 U 型曲线(即所谓“浴缸曲线”)。那是因为刚一投入使用时各部件尚未做到灵活运转,常常容易出问题。经一段运行,便可以稳定下来。而当设备已经历相当时期的运转,便会出现磨损、老化等问题,会使失效率突然提高。这意味着已经到达寿命的终点,即将报废了。软件情况与此不同,它没有 U 型曲线的右半翼,因为它不存在磨损和老化问题。然而软件在投入使用以后,可能要作修改,每次修改

都会引起失效率的提高(参看图 1.1)。事实上,软件在投入运行以后,由于发现错误,为适应运行环境或是需要对其功能加以扩充,都需要对软件进行修改。我们把这种修改称作软件维护。其实,这和硬件的维护有着本质的差别。

图 1.1 软硬件失效率曲线对比

(4) 软件的开发和运行常常受到计算机系统的限制,对计算机系统有着不同程度的依赖性。软件不能完全摆脱硬件单独活动。在开发和运行中必须以硬件提供的条件为依据。有的软件这种依赖性大些,常常为某个型号计算机所专用,这对使用将带来许多不方便。有的软件依赖于某个操作系统。为解除这种依赖性,在软件开发中提出了软件移植的问题。并且把软件的可移植性作为衡量软件质量的因素之一。

(5) 软件的开发至今尚未完全摆脱手工艺的开发方式。软件产品大多是“定做的”,很少能做到利用现成的部件组装成所需的软件。近年来软件技术虽然取得了不少进展,提出了许多新的开发方法,比如充分利用现成软件的复用技术、自动生成技术,也研制了一些有效的软件开发工具或软件开发环境,但在软件项目中采用的比率仍然很低。传统的手工艺开发方式仍然占据统治地位。开

发的效率自然受到很大的限制。对于软件人员来说,开发工作是一种高强度的脑力劳动,没有哪一个软件人员认为,这是一项轻松的工作。

(6) 软件本身是复杂的。有人认为,人类能够创造的最复杂的产物是计算机软件。软件的复杂性可能来自它所反映的实际问题的复杂性,比如,它所反映的自然规律,或是人类社会事务,都具有一定的复杂性;另一方面,也可能来自程序逻辑结构的复杂性。例如,一个系统软件要能处理各种可能出现的情况。软件开发,特别是应用软件的开发常常涉及到其它领域的专门知识,这对软件人员提出了很高的要求。软件的复杂性与软件技术的发展不相适应的状况越来越明显。图 1.2 示出软件技术的发展落后于复杂的软件需求,并且随着时间的推移,这个差距日益加大。

图 1.2 软件技术的发展落后于需求

(7) 软件是相当昂贵的。软件的研制工作需要投入大量的、复杂的、高强度脑力劳动,它的成本自然是较高的。问题不仅于此,值得注意的是硬软件的成本近三十年来发生了戏剧性的变化。无论研制也好,或是向厂家购买也好,在 50 年代末,软件的开销大约占总开销的百分之十几,大部分成本要花在硬件上;但今天这个比例要完全颠倒过来,软件的开销大大超过硬件的开销(参看图 1.3)。美国每年投入软件开发的经费要有几百亿美元。然而,也并非在所有软件开发上的花费都是成功的。

(8) 相当多的软件工作涉及到社会因素。类似于企业管理类

型的软件自然是不言而喻的。许多软件的开发和运行涉及机构、体制及管理方式等问题,甚至涉及到人们的观念和人们的心理。对于这些人的因素重视得不够,常常是软件工作遇到的问题之一。即使是对软件的看法不同也会有很大影响,比如,由于主管部门对正在开发的软件不够理解,因而得不到应有的重视和必要的支持,造成人力和资金上的困难,它直接影响到项目的成败。

图 1.3 计算机系统硬、软件成本比例的变化

以上讨论的是区别于计算机硬件或是其它工程对象,各种软件的共同特点。究竟软件有哪些类型?事实上,要给计算机软件做出科学的分类是很难的,但鉴于不同类型的工程对象,对其进行开发和维护有着不同的要求和处理方法,因此仍然需要对软件的类型给出必要的划分。既然找不到一个统一的严格分类标准,我们从不同的方面去分类是比较符合实际情况的。

(1) 按软件的功能划分:一种通常的分法是三类,即系统软件、支撑软件和应用软件。

系统软件 能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。例如,操作系统、数据库管理系统、驱动程序以及通讯处理程序等。系统软件的工作通常伴随着:频繁地与硬件来往、大量地为用户服务、资源的共享与复杂的进程管理,以及复杂数据结构的处理。系统软件是计算机系统必不可少的一个组成部分。

支撑软件 是协助用户开发软件的软件,包括帮助程序人员开发软件产品,也包括帮助管理人员控制开发的进程。表 1.1 给出了一些支撑软件的实例。

应用软件 能帮助人完成特定领域的工作。现在已经举不出

哪个国民经济部门完全不用计算机。为这些计算机应用领域服务的应用软件种类繁多。其中商用信息处理所占比例是最大的一类，工程和科学计算软件大多属于数值计算问题。值得一提的是，除去那些大量应用的传统领域以外，近年来一些新的应用领域如异军突起，十分引人注目。比如，计算机辅助设计(CAD)、系统模拟、智能产品嵌入软件(如汽车油耗控制、仪表盘数字显示，刹车系统)，以及人工智能软件(如专家系统、模式识别等)。应用软件在这些领域里大显神通，使得传统的产业部门面目一新，带给我们的是惊人的生产效率和巨大的经济效益。

表 1.1 支撑软件举例

一般类型	支持需求分析
文本编辑程序 文件格式化程序 磁盘向磁带作数据传输程序 程序库系统	PSL/PSA 问题描述语言、问题描述 析程序 关系数据库系统 一致性检验程序 CARA 计算机辅助需求分析
支持设计	支持实现
图形软件包 结构化流程图绘图程序 设计分析程序 程序结构图编辑程序	编译程序 交叉编辑程序 预编译程序 联接编辑程序
支持测试	支持管理
静态分析程序 符号执行程序 模拟程序 测试覆盖检验程序	PERT 进度计划评审方法绘图程序 标准检验程序 库管理程序

(2) 按软件的规模划分: 按开发软件所需的人力、时间以及完成源程序的行数可确定六种不同规模的软件, 即微型、小型、中型、大型、甚大型及极大型(参看第八章中表 8. 1)。

(3) 按软件工作方式划分:

- 实时处理软件: 在处理对象所发生不断变化的活动中, 能随时配合完成量测、分析和控制的软件称为实时处理软件。对这样的软件来说, 处理的时间是被严格限定的, 如果在任何时间超出了这一限制, 都将造成事故。

- 交互式工作的软件接收用户给出的信息, 不过时间上没有严格的限定。这种工作方式给用户很大的灵活。近年来, 终端设备更加普及, 交互式软件到处可见。一个重要的问题日益显得突出, 这就是交互式软件的用户接口设计。良好的用户接口设计将给用户带来极大的方便。

- 分时工作方式是按固定的时间段, 轮流处理多个任务。

- 批处理方式则是把多个任务一起送入机器, 按顺序逐个处理的传统工作方式。

(4) 按软件服务对象的范围划分:

有的软件只供一个(或少数几个)用户使用, 可称这样的软件为项目软件(project software)。例如, 军用防空指挥系统、卫星轨道控制系统的软件属于这一类。这类项目软件中有的软件, 其开发目的带有试验研究性质。项目完成后可能需要在在此基础上做进一步开发工作。

另一类则是要提供市场或是为千百个用户服务, 我们称为产品软件(Product software)。一些通用的软件, 如绘图软件包、编译程序、文本处理程序、报表生成程序等都属于这一类。由于要参与市场竞争, 这类软件的功能、性能如何是很重要的因素。

(5) 按软件的使用频度划分:

有的软件开发出来仅供一次使用, 比如人口普查、工业普查所需软件。若干年才进行一次普查, 前些年开发的软件在若干年后很难适用。有的统计资料或试验数据需按年度做统计分析, 相应的软件每年运行一次。另外一些问题, 如天气预报, 需每天进行数据的及时处理, 相比之下, 这类软件具有较高使用频度, 显然, 开发不同使用频度的软件, 有不同的要求, 不可一律看待。

(6) 按软件的失效影响划分:

工作在不同领域的软件, 适应其不同的需求, 在运行中对可靠性也有不同的要求。有的软件如果在工作中出现了故障, 造成软件失效, 可能给整个系统带来不大的影响。比如可能带来一些不便, 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。比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。开发这类软件自然应从多个方面采取措施, 确保质量, 做到万无一失。

1.2 软件的发展和软件危机

本世纪 40 年代中出现了世界上第一台计算机以后, 就有了程序的概念。可以认为它是软件的前身。经历了 30 年的发展, 使我们得以对软件有了更为深刻的认识。在 30 年中, 计算机软件经历了三个发展阶段:

- 程序设计阶段, 约为 50 至 60 年代
- 程序系统阶段, 约为 60 至 70 年代
- 软件工程阶段, 约为 70 至 80 年代

从表 1.2 中我们可以看到三个发展时期主要特征的对比。30 年来, 最根本的变化体现在:

(1) 人们改变了对软件的看法。在 50 年代到 60 年代, 程序设计曾经被看作是一种任人发挥创造才能的技术领域。当时人们认为, 写出的程序只要能在计算机上得出正确的结果, 程序的写法可以不受任何约束。一些程序尽管很难被别人看懂, 但仍然认为, 只

表 1.2 计算机软件发展的三个时期及其特点

特 时 期 点	程 序 设 计	程 序 系 统	软 件 工 程
软件所指	程 序	程序及说明书	(项目软件) 产 品 软 件
主要程序设计语言	汇编及机器语言	高 级 语 言	高级语言、系统 程序设计语言
软件工作范围	程序编写	包括了设计 和测试	软件生存期
需求者	程序设计者本人	少 数 用 户	市 场 用 户
开发软件组织	个 人	开 发 小 组	开 发 小 组 软 件 工 厂
软件规模	小 型	中 小 型	大 中 (小) 型
决定软件质量 的因素	个人程序技术	小组技术水平	管 理 水 平
开发技术和手段	子 程 序 程 序 库	结构化程序 设计	数据库、开发工 具开发环境、工 程化开发方法、 标准和规范
维护责任者	程序设计者	开 发 小 组	专 职 维 护 人 员
硬件特征	价 高 存储容量小 工作可靠性差	降价, 速度、 容量及可靠性 有明显提高	向超高速、大容 量及微型化发展
软件特征	完全不受重视	软件技术的 发展不能满 足需要, 出现 软件危机	开发技术有了前 进, 但未获突破 性进展, 价高 软件危机并未完 全摆脱

有通篇充满了程序技巧, 使用了许多窍门的程序才是高水平的好

程序。然而,随着计算机的广泛使用,人们逐渐抛弃了这种观点。因为,对于小的程序,仅供极小范围使用(例如只是程序设计者本人或只有几个人),尚可“孤芳自赏”。对于稍大的程序,并需要较长时间为许多人使用的程序,情况就完全同了。人们要求这些程序要容易看懂、容易使用并且要容易修改和扩充。于是程序便从个人按自己意图创造的“艺术品”转变为能被广大用户接受的工程化产品。这时程序中难于理解的技巧成了有害的东西。

(2) 软件的需求是软件发展的动力。早期的程序开发者只是为了满足自己的需要,这种自给自足的生产方式仍然是其低级阶段的表现。进入软件工程阶段以后,软件开发的成果具有社会属性,它要在市场中流通以满足广大用户的需要。软件开发者和用户的分工和责任也是十分清楚的。

(3) 软件工作的范围从只考虑程序的编写到涉及整个软件生存期。关于软件生存期的概念将在下面一节介绍。

在软件技术发展的第二阶段,随着计算机硬件技术的进步,计算机的容量、速度和可靠性有明显提高,生产硬件的成本降低了。计算机价格的下跌为它的广泛应用创造了极好的条件。在这一形势下,要求软件能与之相适应。一些开发复杂的、大型的软件项目提了出来。然而软件技术的进步一直未能满足形势发展提出的要求。在软件开发中遇到的问题找不到解决的办法,致使问题积累起来,形成了日益尖锐的矛盾。这些问题归结起来有:

(1) 由于缺乏软件开发的经验和关于软件开发数据的积累,使得开发工作的计划很难制定。主观盲目的制定计划,执行起来和实际情况有很大差距。致使经费预算常常突破。对于工作量估计不准确,进度计划无法遵循,开发工作完成的期限一拖再拖。已经拖延了的项目,为了加快进度赶上去而增加人力,结果适得其反,不仅未能加快,反面更加延误了。在这种情况下,软件开发的投资者和软件的用户对软件开发工作既不满意,也不信任。