



中等职业学校电子信息类教材 计算机技术专业

软件工程

谢夫娜 丁兆海 主编
王 森 主审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

本书配有电子教学参考资料包

<http://www.phei.com.cn>

中国计算机学会推荐教材 高等学校计算机专业系列教材

清华大学出版社

软件工程

第二版

清华大学出版社

ISBN 7-302-11111-1

中等职业学校电子信息类教材（计算机技术专业）

软 件 工 程

谢夫娜 丁兆海 主编

王 森 主审

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书按照软件生存周期和常用的软件开发模型,较全面、系统地介绍了有关软件工程的基本知识,内容包括可行性研究与计划、需求分析、软件设计、软件编码、软件测试、软件维护、软件项目管理等。为适应计算机技术不断发展和不同教学的需求,推广新技术,本书还增加了 CMM 模型、ISO 9000 系列国际标准、面向对象的方法学等选学内容,突出教材的适用性和先进性。

本书针对中等职业教育层次,特别注重基本概念、基本方法和基本技术的介绍和实践技能的训练。叙述语言精练,取材难易适中,并有丰富的应用实例和各阶段的文档编制指南,实用性很强。书中每章都有小结和习题,便于教学和自学。

本书除可作为中等职业学校计算机技术专业的专业教材外,还可作为对软件工程知识感兴趣的自学者的参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

软件工程/谢夫娜主编. —北京:电子工业出版社, 2004.6
中等职业学校电子信息类教材. 计算机技术专业
ISBN 7-5053-9969-1

I. 软… II. 谢… III. 软件工程—专业学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2004)第 045811 号

责任编辑:徐晓光

印 刷:北京季蜂印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:10 字数:256 千字

印 次:2004 年 6 月第 1 次印刷

印 数:5 000 册 定价:13.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。



随着现代信息技术的不断发展,中等职业教育中的计算机技术专业教学内容也越来越丰富,为了满足社会劳动岗位的需求,学生掌握的知识也必须越来越宽广。目前我国IT业发展迅速,软件开发领域的“蓝领”极其缺乏,中等职业学校的毕业生要想进入软件开发的行业,掌握计算机软件开发的过程、方法及工具的基本知识是十分必要的。

编写思路:本教材的任务是使学生掌握有关软件工程的基本知识,初步掌握软件开发的基本思想、方法和相关技能,为学生适应职业岗位的需要打下一定的基础。因此,教材的主线是重视基本知识和基本概念的传授,按照我国目前执行的软件开发国家标准进行组织教材内容,注意理论与实践的结合。

本书在编写的过程中,针对中等职业教育的特点,努力做到以下几点:

(1) 传统理论与先进知识的结合。按照软件生存周期模型全面、系统地介绍了软件工程的基本概念、基本技术和基本方法;突出实用性,结合具体实例分析如何规范地进行软件开发。考虑到计算机软件技术正在飞速发展,在本书的最后概述了软件工程的最新技术,以便读者进一步学习和提高。

(2) 课程的科学体系与实用性的结合。在讲授方法上注意由浅入深,由表及里。叙述语言精练,并有丰富的应用实例。在每章后面对本章内容进行小结。每章最后附有代表性的习题,对巩固所学知识很有帮助。

(3) 采用模块化、单元化的组织方式进行教材编写和实施教学。突出“浅显”、“够用”的原则,把握好“必须”和“足够”这两个“度”。

(4) 突出基础性和实用性,讲述最基本、最常用的知识和最基本的实践技能。在文字叙述上力求条理清晰、简明易懂、用例恰当、便于自学。

(5) 为适应国情和不同的需求,推行新技术,增加了CMM模型、ISO 9000系列国际标准、面向对象的方法学等选学内容,突出教材的适用性和先进性。

“软件工程”在中等职业学校的教学中是一个新的课题,原有的同类教材大多是针对单一的软件应用或某些专业基本知识培养的相关内容。通过本教材的学习会使学生系统掌握软件开发全过程所必须具有的知识,掌握软件项目开发和管理的实践技能。

目前市场上常见的“软件工程”用书,大多为高等学校教材,内容偏多、偏难。适合中等职业学校教学的教材还不多见。本书力求达到帮助读者理解问题的目的。由于软件工程学科发展太快、概念不断更新,书中内容难免有所疏漏,恳切希望广大读者批评指正,不胜感谢。为了方便教师教学,本书还配有教学指南、电子教案及习题答案(电子版),请有此需要的教师与电子工业出版社联系,我们将免费提供。E-mail:ve@phei.com.cn

编 者
2003年12月





第 1 章 软件工程概述	1
1.1 软件的特点与分类	1
1.1.1 软件的概念与特点	1
1.1.2 软件的分​​类	2
1.2 软件危机	3
1.2.1 软件开发技术的发展过程	3
1.2.2 软件危机	3
1.2.3 产生软件危机的原因	4
1.2.4 如何解决软件危机	5
1.3 软件工程过程与软件生存周期	5
1.3.1 什么是软件工程	5
1.3.2 软件工程过程	6
1.3.3 软件生存周期	6
1.4 软件开发模型	8
1.4.1 瀑布模型 (Waterfall Model)	9
1.4.2 快速原型化模型	10
1.4.3 其他开发模型	12
1.5 软件工程的基本原理	13
1.5.1 软件工程的基本目标	13
1.5.2 软件工程的原理	13
本章小结	14
习题 1	14
第 2 章 软件需求分析	16
2.1 可行性研究	16
2.1.1 可行性研究的内容	16
2.1.2 可行性研究报告的内容	16
2.2 需求分析	17
2.2.1 进行需求分析的原因	17
2.2.2 需求分析定义	17
2.3 需求分析的内容	18
2.3.1 需求分析的过程	18
2.3.2 需求获取的方法	19

2.4	结构化分析方法	20
2.4.1	结构化分析模型的结构	20
2.4.2	实体—关系图 (ERD)	20
2.4.3	数据流图 (DFD)	21
2.4.4	状态迁移图 (STD)	25
2.4.5	数据词典	26
2.4.6	加工逻辑说明	27
2.5	需求分析的控制技术	28
2.5.1	有限状态机	29
2.5.2	Petri 网	29
2.6	软件需求规格说明和需求评审	29
2.6.1	软件需求规格说明和初步用户手册	29
2.6.2	软件需求评审	33
	本章小结	35
	习题 2	35
第 3 章	软件设计	39
3.1	软件设计的基本概念	39
3.1.1	什么是软件设计	39
3.1.2	如何将分析模型转换为软件结构设计	39
3.1.3	软件设计在软件开发中的地位	39
3.1.4	软件设计方法的分类	40
3.2	软件设计原则——模块独立性	40
3.2.1	模块独立性原则的含义	40
3.2.2	模块的耦合性	41
3.2.3	模块的聚合性	43
3.2.4	模块规模应该适中	45
3.3	软件设计的其他原则	46
3.3.1	信息隐蔽和局部化的原则	46
3.3.2	抽象的原则	47
3.3.3	控制层次适中的原则	47
3.4	概要设计	48
3.4.1	概要设计的过程	49
3.4.2	概要设计使用的图形工具	49
3.4.3	面向数据流的设计方法	50
3.5	详细设计	52
3.5.1	详细设计的任务	52
3.5.2	详细设计使用的图形工具	53
3.5.3	面向数据结构的设计方法	56
3.6	其他软件开发方法简介	60

3.6.1	面向问题的分析法	60
3.6.2	原型化方法	61
3.6.3	面向对象的软件开发方法	61
3.6.4	可视化开发方法	61
3.7	设计规格说明书与设计评审	62
3.7.1	软件概要设计说明书的大纲	62
3.7.2	软件详细设计说明书的大纲	64
3.7.3	设计评审	65
本章小结		65
习题 3		66
第 4 章	程序编码	68
4.1	程序设计概述	68
4.1.1	程序设计的发展	68
4.1.2	程序设计语言特点	69
4.2	结构化程序设计	69
4.2.1	自顶向下的程序设计方法	69
4.2.2	结构化设计的原则	70
4.3	程序设计的风格	70
4.3.1	书写程序的风格	70
4.3.2	程序效率	71
4.4	程序设计的复杂性度量	72
本章小结		72
习题 4		72
第 5 章	软件测试	74
5.1	基本概念	74
5.1.1	软件测试的定义	74
5.1.2	软件测试的基本原则	75
5.1.3	软件测试的分类	76
5.1.4	软件测试的信息流	77
5.2	软件测试的方法	77
5.2.1	黑盒测试	78
5.2.2	白盒测试	78
5.3	测试用例的设计	79
5.3.1	逻辑覆盖	79
5.3.2	等价类划分	84
5.3.3	边界值分析法	85
5.3.4	错误推测法	86
5.3.5	因果图法	86
5.4	软件测试的步骤	88

5.4.1	单元测试	89
5.4.2	集成测试	90
5.4.3	确认测试	93
5.4.4	系统测试	94
5.5	测试工具	95
5.5.1	人工测试	95
5.5.2	测试工具的分类	96
5.5.3	测试工具的选择	97
5.6	调试	97
5.6.1	调试的步骤	98
5.6.2	调试的策略	98
5.6.3	调试的原则	99
5.7	软件的可靠性	99
5.7.1	软件可靠性的定义	99
5.7.2	程序的正确性证明	100
本章小结		100
习题 5		101
第 6 章	软件维护	104
6.1	软件维护的概念	104
6.1.1	软件维护的含义	104
6.1.2	影响维护工作量的因素	105
6.2	软件维护的特点	107
6.2.1	软件工程与软件维护的关系	107
6.2.2	维护成本	107
6.2.3	软件维护的内容	108
6.3	软件维护的过程	108
6.3.1	软件维护机构	108
6.3.2	软件维护申请报告	109
6.3.3	软件维护的工作流程	109
6.3.4	软件维护的实施	110
6.3.5	软件维护文档	111
6.3.6	软件维护的评价	111
6.4	软件的可维护性	112
6.4.1	可维护性的地位	112
6.4.2	软件的可维护性度量	112
6.4.3	提高软件可维护性的方法	112
6.5	软件维护的副作用	113
6.5.1	程序修改的步骤	113
6.5.2	修改程序的副作用	114

6.6 软件再生工程	115
本章小结	116
习题 6	116
第 7 章 软件项目管理	119
7.1 软件项目管理过程	119
7.1.1 什么是软件项目管理	119
7.1.2 软件项目管理的过程	120
7.2 软件的生产率和质量的度量	121
7.2.1 如何评价软件的质量	121
7.2.2 影响软件生产率的因素	123
7.3 软件项目的估算	123
7.3.1 对估算的看法	123
7.3.2 软件项目计划的目标	124
7.3.3 软件规模是项目估算的基础	124
7.3.4 软件项目规模估算方法	124
7.3.5 软件开发中的资源	126
7.4 软件开发成本估算	127
7.4.1 软件开发成本估算方法	127
7.4.2 软件开发成本估算的经验模型	129
7.5 软件项目进度安排	130
7.5.1 进度安排中应考虑的几个问题	130
7.5.2 进度安排的方法	131
7.6 软件风险管理	131
7.6.1 风险的分类	132
7.6.2 风险的识别	132
7.6.3 风险评估	132
7.6.4 风险的评价	132
7.6.5 风险的驾驭和监控	132
7.7 软件项目的组织	133
7.7.1 软件项目管理的组织模式	133
7.7.2 人员组织与管理	133
7.8 软件配置管理	134
7.8.1 软件配置管理人员应完成的主要任务	135
7.8.2 软件配置管理应提供的功能	136
7.9 软件过程能力评估	136
7.9.1 国际标准 ISO 9000	136
7.9.2 软件过程能力评估的 CMM 模型	137
本章小结	138
习题 7	138

第 8 章 面向对象的技术和应用 140

8.1 面向对象的基本知识 140

8.1.1 面向对象的概念 140

8.1.2 面向对象方法及优点 142

8.2 面向对象的分析方法 142

8.2.1 面向对象的系统分析 142

8.2.2 如何进行对象的划分 143

8.2.3 制定协议 144

8.2.4 逐步求精 145

8.3 统一建模语言 145

8.3.1 统一建模语言简介 145

8.3.2 统一建模语言的内容 146

本章小结 147

习题 8..... 147

参考文献 148

第1章 软件工程概述



本章知识要点：本章主要讲述了软件的概念和特点及软件危机产生的原因、软件生存周期及两种软件开发模型、软件工程过程和软件工程的原理等方面的内容。

- ◆ 软件的定义
- ◆ 软件的发展与软件危机
- ◆ 软件生存周期与软件开发模型
- ◆ 软件工程基本原理

随着计算机技术的飞速发展，人类社会已跨入信息时代，计算机系统成为信息时代最重要的工具，计算机应用则成为信息时代的主要特征。

任何计算机系统都是由硬件和软件两大部分组成，计算机硬件发展的特点是功能越来越强、性能质量越来越高、价格却越来越低；而计算机软件发展的趋势却是规模和范围越来越广、开发难度和代价越来越大、质量保证与生产效率却越来越难以达到人们的需要。

计算机软件发展已成为计算机系统发展的关键，为了寻求出路，人们从 20 世纪 60 年代后期开始，提出用“工程”的概念来定义软件的开发、运行与维护，此后便逐步形成了一门新兴学科——软件工程学。

1.1 软件的特点与分类

1.1.1 软件的概念与特点

1. 什么是软件

在 20 世纪 50 年代，软件伴随着第一台电子计算机的问世诞生了。以写软件为职业的人也开始出现，他们大多是经过训练的数学家和电子工程师。20 世纪 60 年代美国大学开始授予计算机专业的学位，教人们编写计算机软件。20 世纪中期软件产业从零开始起步，在短短的 50 多年时间里迅速发展成为推动人类社会发展的龙头产业，并造就了一批百万、亿万富翁。随着信息产业的发展，软件对人类社会越来越重要。

那么，如何给软件定义呢？软件是什么？软件一词的英文是 software，现在普遍接受的软件的定义是软件是计算机系统中与硬件(Hardware)相互依存的另一部分，它包括程序(Program)、相关数据(Data)及其说明文档(Document)。该定义将软件的组成成分分成两个部分，一部分是指在计算机中可执行的程序和相关的数据库，另一部分是指与软件开发、使用和维护相关的所有文档资料。



注意

理解软件的定义我们要注意以下几点:

- (1) 程序是能够实现一定功能的指令序列。
- (2) 数据是使程序能正常操纵信息的数据结构。
- (3) 文档是与软件开发、维护和使用有关的图文资料。

2. 软件的主要特点

在现代计算机系统中必须装载软件才能发挥作用, 我们知道, 计算机的纯硬件部分通常统称为裸机。一台裸机即使有很强的功能, 有上亿次的运算速度, 但是, 如果没有软件, 你就根本无法使用它。软件有如下特点:

(1) 软件的生产大大难于硬件的生产。软件产品主要是程序编码, 编码完成以前, 既看不到实体, 也无法衡量其质量和效率, 完全依赖于工作人员的素质(品德及业务能力)。因此, 管理和控制软件的生产过程要比硬件生产管理和控制过程难得多。

(2) 软件的规模都很庞大。软件的规模越大其复杂程度就越高, 涉及的技术问题就越广泛, 需要的开发人员就会越多, 生产成本也就越大。

(3) 软件产品扩大生产容易, 且经久耐用。硬件产品试制成功后, 要想投入批量生产会遇到人力、物力和资金等方面的许多困难, 而且硬件产品在使用过程中容易磨损和老化。软件产品在开发成功后, 只需对原版软件进行复制便可扩大再生产, 非常容易和方便。并且, 软件产品不会因使用时间过长而被磨损和用坏。

(4) 软件的维护修改复杂。软件在交付使用以后, 为了纠正原有的错误和适应用户新的要求以及满足硬件环境的变化, 都需要对软件进行维护和修改, 但是每次修改都可能引入新的错误, 因此, 软件的维护要比硬件维护复杂得多。

1.1.2 软件分类

软件工程的对象不同, 对其进行开发和维护就会有不同的处理方法, 因此, 对软件类型进行科学的划分是很必要的。从不同的方面划分, 会有如下一些类型。

(1) 按照软件的功能, 一般可以分为以下几种类型。

① 系统软件。如大家熟悉的 Windows 系列、UNIX 以及高级语言的编译系统, 设备驱动程序等。系统软件的作用是控制和管理计算机系统内的各种资源, 提供系统服务, 方便用户使用。

② 应用软件。如 Word、CAD、专家系统、事物管理等。应用软件是为特定目的服务的一类软件。

(2) 按照软件的工作方式, 一般可分为以下几种类型。

① 实时处理软件。在计算机的某些应用领域内, 要求对实时采样数据立即处理并做出及时的反应, 如果超出限定的时间就可能引起信息的丢失。例如卫星发射过程中, 必须对出现的各种情况立即进行分析和处理。

② 分时软件。我们知道, 为了提高资源利用率采用了并行操作的技术, 分时主要是指若干并发程序对时间的共享。分时软件响应快, 界面友好, 便于资源共享。

③ 批处理软件。早期的计算机系统大多数采用批处理系统。批处理软件是把一组作业以



成批的方式一次运行，按顺序逐个处理软件。

1.2 软件危机

1.2.1 软件开发技术的发展过程

软件对于人类而言是一个全新的东西，其发展历史不过四五十年。人们对软件的认识经历了一个从浅到深的过程。从第一台电子计算机问世以来，每次硬件技术的突破，都为软件技术的新发展提供了更为广阔的空间，人们对计算机软件的观点及目标等方面都发生了很大的变化。软件的发展大体上经历了以下四个阶段。

1. 机器码编程阶段

机器码编程阶段，也称为程序设计的原始时期。这一时期计算机的用途主要是科学计算或与军事有关的问题计算，既没有汇编语言，也没有高级语言，程序人员编写机器能识别的机器码程序，程序的质量完全依赖于程序员个人的技巧。

2. 符号语言编程阶段

符号语言编程阶段，也称为基本软件时期。这个阶段出现了面向机器的汇编语言和面向应用的各种高级语言。这一阶段计算机应用范围大大扩大，编程的目的是要确定程序人员定义的一系列计算机指令执行序列。同时高级语言的发展，使用高级语言编写的程序与执行程序的计算机硬件无关。人们意识到大型软件系统的开发工作与人们早已熟知的建设工程极为类似。

3. 提出“程序=数据结构+算法”阶段

提出“程序=数据结构+算法”阶段，也称为程序设计方法时期。这一阶段开始提出结构化程序设计方法和模块化程序设计方法，软件的开发费用也急剧上升。

4. 软件工程时期

随着软件规模的扩大，开发软件系统的重点开始转向构造系统的方法。这一时期对软件系统的生命周期有了新的认识。软件开发的新技术得到了广泛的应用，由于 Internet 技术的高速发展，智能软件、并行软件与 Internet 环境下的分布式软件开发技术出现了新的课题。

注意

以上是按时间顺序划分软件开发的各个阶段，从开发方法上，可以分为以下几个阶段。

(1) 个体开发阶段。自己开发自己应用，开发软件就是写程序，不存在系统化的开发方法。

(2) “软件作坊”阶段。软件开发向多人合作的方向发展，软件生产出现了专营单位。

(3) 软件工程阶段。人们为了克服软件危机，提出用工程的观点对付软件的开发。

(4) 新技术应用阶段。软件的形式化开发方法和自动生成技术等都在不断地发展之中。

1.2.2 软件危机

1. 定义

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)这个名词。软件危机是指在计算机软件的开发和维护过程



中所遇到的一系列严重问题。这些问题并不局限于那些被认为是“未能正常运行的软件”。在软件技术发展的过程中,出现了很多的问题,形成尖锐的矛盾,最终导致了软件危机。“软件危机”使得人们开始对软件及其特性进行更深一步的研究,从而改变了早期对软件的不正确看法。早期那些被认为是优秀的程序常常很难被别人看懂,通篇充满了程序技巧。现在人们普遍认为优秀的程序除了功能正确、性能优良之外,还应该容易看懂、容易使用、容易修改和扩充。

2. 软件危机的具体表现

(1) 软件开发计划难以制定。在软件开发的过程中,由于多方面的原因,无法遵循软件的进度计划,以致超出预先的软件开发成本估算,使软件的交工期限一再拖延。

(2) 软件的需求模糊使得软件需求分析过程中交换意见的途径经常阻塞,分析员和用户之间无法建立正常的工作关系,在这种情况下,程序员仓促编写程序,导致最终的软件产品不符合用户的需求。

(3) 软件的可维护性差。由于未采用先进的软件工程方法编制程序,导致程序中的错误难以测试、难以改正,不能根据数据环境的变化来修改软件以适应用户的需求,或者软件在使用过程中不能根据用户的要求改进软件现有的功能,增加新的功能。

(4) 软件文档资料不全。我们所设计的软件除了程序之外,还有很重要的一部分,那就是文档。软件开发过程中缺乏统一的规范指导,设计人员各行其事,缺乏完整的文档说明,这样的软件如果出现了问题,难以维护。

(5) 难以控制开发进度。软件开发过程中遇到的各种意想不到的情况层出不穷,同样的软件算法,可以用不同的描述形式来实现。为完成一个庞大的软件,常需要建立复杂的逻辑体系。当我们投入资金后,能否得出结果,得出怎样的结果,事先难以预料,这些特点给软件开发进度的控制带来了很大的困难。

1.2.3 产生软件危机的原因

由于软件本身的特点以及采用了不正确的软件开发方法,致使软件开发和维护过程中存在很多的问题。产生软件危机的部分原因如下。

1. 软件规模庞大

假设一人一年可以开发出一个 10 000 行的程序,为了开发一个 5 000 万行的目标代码,是否集中 5 000 人的力量一年可以完成该软件的开发呢?答案是不可能,因为如何保证每个人开发的软件代码合在一起构成一个高质量的大型软件系统,是一个极端复杂的问题。软件设计的过程中要充分分析软件的体系结构,模块之间的关系,构件的复用,以及如何保证质量保证体系的实施,编写符合规则的文档等。软件复杂性的提高决定了需要很长的研制周期和大量人力、物力的投入。

2. 开发人员的错误观念

在开发人员身上出现的错误常常表现在多个方面:如需求分析不够完善,没有把用户需求弄清就先做起来再说;不重视文档资料的作用,只重视程序编码和个人“技艺”;缺乏对大型软件的开发与管理经验和通力合作的精神,易造成接口部分的疏漏和错误等等。



1.2.4 如何解决软件危机

首先,必须借鉴人类长期以来从事各种工程项目的经验,吸取几十年来人们从事计算机软硬件研究和开发的有效的原理、技术和方法,形成一个组织良好、管理严密的工程项目团体。

其次,应该开发出能够减轻人类烦琐重复工作的软件工具,把各阶段的工具有机地结合在一起,支持软件开发的全过程。

1.3 软件 Engineering 过程与软件生存周期

从以上软件危机的表现以及软件危机原因的产生可以看出,解决软件危机不是一个简单的问题,我们应该排除传统观念的束缚和某些错误的认识,利用软件工程的观点开发软件。实践经验表明,按工程化的原则和方法组织软件开发是行之有效的,也是解决软件危机的一个主要途径。

1.3.1 什么是软件工程

1. 定义

1968年秋季,NATO(北约)的科技委员会召集了近50名一流的编程人员、计算机科学家和工业界巨头,讨论和制定摆脱“软件危机”的对策。在那次会议上第一次提出了软件工程(Software Engineering)这个概念。到20世纪末,软件工程整整走过了30多年的历程。在这30多年的发展中,人们针对软件危机的表现和原因,经过不断的实践和总结,逐渐认识到:按照工程化的原则和方法组织软件开发工作,是摆脱软件危机的一个主要出路。尽管“软件危机”并未被彻底解决,但软件工程30多年的发展仍可以说是硕果累累。下面我们给出一个软件工程的定义,然后简单讨论一下软件工程所包括的内容。

软件工程是软件开发、运行、维护和引退的系统方法。软件工程是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列完善的工程化原则。软件工程包括两方面内容:软件开发技术和软件项目管理。软件开发技术包括软件开发方法学、软件工具和软件工程环境;软件项目管理包括软件度量、项目估算、进度控制、人员组织、配置管理、项目计划等。

统计数据表明,大多数软件开发项目的失败,并不是由于软件开发技术方面的原因。它们的失败是由于不适当的管理造成的。遗憾的是,尽管人们对软件项目管理重要性的认识有所提高,但在软件管理方面的进步远比在设计方法学和实现方法学上的进步小,至今还提不出完整的管理软件开发的通用指导原则。

2. 软件工程三要素

(1) 方法。软件工程方法包括多方面的任务,如项目计划、需求分析、结构设计、过程设计、程序设计、软件测试、软件维护等。它为软件开发提供了“如何做”的技术。

(2) 工具。在软件开发的每个阶段都有许多重复的工作需要做,在适当的软件工具辅助下,开发人员可以把这类工作做得既快又好,如果把各阶段使用的软件工具有机地结合成一个整体,支持软件开发的全过程,则称为软件工程支撑环境。软件工具集成起来,形成CASE(Computer Aided Software Engineering)计算机辅助软件工程,它是指人们在计算机的辅助



下,从事软件工程活动,即支持软件的开发、运行、维护和管理的全过程。

(3) 过程。过程定义了方法使用的先后顺序、软件开发各阶段的文档资料,质量保证的实施以及软件开发各阶段的里程碑。

1.3.2 软件工程过程

软件工程过程 (Software Engineering Process)是在软件工具支持下由软件人员完成的一系列软件工程活动。事实上,软件工程过程是一个软件开发机构针对某类软件产品为自己规定的工作步骤。

注意

软件开发过程是把用户的要求转变成软件产品的过程。此过程包括需求分析,把需求转换为设计,再把设计转变为代码,最后进行软件测试和交付使用。

1.3.3 软件生存周期

1. 定义

软件生存周期,有时也称为软件生命周期,它是指软件的孕育、诞生、成长、成熟、衰亡的生存过程,即从开始考虑其软件产品到该软件产品不能再使用的整个时期。它从时间的角度对软件开发和维护的整个过程进行分解,划分为各个阶段,每个阶段的任务相对独立,前一阶段任务的完成是后一阶段工作开始的前提和基础,每一阶段结束之前,必须进行严格的技术评审,通过之后这一阶段才算结束,从而使软件开发的整个过程有效地进行下去,既能保证软件的质量又能提高软件的可靠性和可维护性。

注意

软件开发周期是与软件生存周期非常接近的概念,它是指从开始考虑软件产品到该软件产品交付使用为止的整个时期。

2. 软件生存周期阶段的划分

目前划分软件生存周期阶段的方法有很多种,但必须遵循一条基本原则,即各阶段的任务尽可能独立,降低各阶段任务的复杂程度,简化各阶段之间的联系,下面简要介绍软件生存周期各个阶段的基本任务。

粗略划分,软件生存周期由计划、开发和运行三个时期组成,每一时期可再细分一些工作阶段。按照我国国家标准《计算机软件开发规范》(GB8566-88)把软件生存周期划分为8个阶段,分别简述如下:

(1) 计划时期。这是开始时期,主要任务是确定软件开发的总目标,确定是否存在可行的系统解决方案。确认问题有解后,制定开发工程的资源及进度计划。该时期的工作阶段称为可行性研究与计划。一般做法是先给问题定义,弄清要解决的问题是什么,明确问题的性质、规模和工程目标,然后进行可行性研究,经过调查访问确定问题是否有解。若有解,经粗略的分析设计得出不同的系统方案,并对其在技术上、经济上、社会因素上的可行性进行研究,得出能不能做和该不该做的结论。根据结论制定开发计划,写出可行性研究报告。

(2) 开发时期。在开发时期,通过分析和设计,最终实现软件系统,这个时期可细分为六个阶段。