

# 软件工程基础

编著 赵一丁

北京邮电大学出版社

·北京·

## 内 容 简 介

本书全面系统地阐述了自动化领域软件工程的基本概念、基本技术和基本方法。全书共 16 章, 主要内容包括: 软件开发过程和软件开发模型、软件需求分析、软件测试和软件维护、软件工程管理、软件设计基础、UML 建模语言、实时系统软件设计、工业自动化组态软件技术。组态软件技术的内容包括实时数据库设计和变量组态、图形界面设计、报表组态、设备驱动与管理、通讯组态、工控组态软件设计应用实例。附录包括软件文档格式、常用工控组态软件(力控 PCAuto、西门子 Win CC 和 MCGS)简介。

本书内容丰富, 图文并茂, 通俗易懂。本书可作为高等院校本科及专科自动化专业、测控专业、自控专业、机电专业、计算机相关专业的教材, 也可供软件工程技术人员参考。

## 图书在版编目(CIP)数据

软件工程基础/赵一丁编著. —北京: 北京邮电大学出版社, 2006

ISBN 7-5635-1277-2

I. 软... II. 赵... III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2006)第 084839 号

---

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(100876)

北方营销中心: 电话: 010-62282185 传真: 010-62283578

南方营销中心: 电话: 010-62282902 传真: 010-62282735

E - mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷:

开 本: 787 mm×1 092 mm 1/16

印 张: 18.75

字 数: 439 千字

印 数: 1—3 000 册

版 次: 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

---

ISBN 7-5635-1277-2/TP·238

定价: 28.00 元

· 如有印装质量问题请与北京邮电大学出版社营销中心联系 ·

## 高等院校自动化新编系列教材

### 编委会

主任 汪晋宽

副主任 金海明 罗云林 张美金 崔光照

委员 (排名不分先后)

于丁文 王凤文 王建国 马淑华 石云霞

齐世清 任彦硕 张家生 张健 杨建忠

柳明丽 罗长杰 金伟 赵宏才 赵一丁

顾德英 舒冬梅 藏小杰 郑安平

秘书 顾德英(兼) 马淑华(兼)

## 编写说明

一本好的教材和一本好的书不同,一本好的书在于其内容的吸引力和情节的魅力,而一本好的教材不仅要对所介绍的科学知识表达清楚、准确,更重要的是在写作手法上能站在读者的立场上,帮助读者对教材的理解,形成知识链条,进而学会举一反三。基于这种考虑,在充分理解自动化专业培养目标和人才需求的前提下,我们规划了这套《高等院校自动化新编系列教材》。

本套系列教材共包括 21 册,在内容取舍划分上,认真分析了各门课程内容的相互关系和衔接,避免了不必要的重复,增加了一些新的内容。在知识结构设计上,保证专业知识完整性的同时,考虑了学生综合能力的培养,并为学生继续学习留有空间。在课程体系规划上,注意了前后知识的贯通,尽可能做到先开的课程为后续的课程提供基础和帮助,后续的课程为先开的课程提供应用的案例,以便于学生对自动化专业的理解。

《高等院校自动化新编系列教材》编委会

2005 年 8 月

# 前 言

软件工程作为一门有关软件开发的工程方法学,在软件开发中的指导意义与基础地位已经越来越多地得到了信息业界和自动化业界的高度重视。软件工程已经成为计算机及其相关专业的专业核心课程。

在自动化领域,随着嵌入式实时系统、工业控制计算机、PLC 和工业网络通讯应用的日益扩大,软件开发的规模也越来越大,对软件的可靠性和可维护性的要求也越来越高。采用软件工程的理论、方法和技术进行软件开发和软件项目管理,对自动化领域的软件开发同样是很重要的。

自动化领域的软件开发有其自身的特点。一是编程语言和开发工具多种多样。嵌入式系统的编程语言有汇编语言、C 和 C++ 语言、实时 Java (J2ME);在通用 PC 计算机上的编程语言有 C++、Java、C#、Visual Basic 等;在 PLC 中采用的编程语言有梯形图、指令表等。开发工具有编程语言的开发工具和工控组态软件开发平台。二是对软件的可靠性要求很高。在许多自动控制系统中,软件缺陷可能会导致严重后果。三是自动化系统往往要与许多硬件设备连接,软件调试和软件测试更加复杂和困难。

在自动化领域,采用组态控制技术,以组态软件为软件平台进行软件开发越来越受到用户的青睐。组态软件能以灵活多样的组态方式(而不是编程方式)提供良好的用户开发界面和简捷的使用方法,其预先设置的各种软件模块可以非常容易地实现和完成工控层的各项功能,并能同时支持各种硬件厂家的计算机和 I/O 设备。组态软件与其支持的硬件设备一起,为用户提供自动化系统的软硬件整体解决方案。嵌入式软件开发也是组态控制技术的应用的领域。

本书介绍了软件工程的基础知识、基本理论、基本方法和基本技术,重点介绍了自动化领域软件的开发方法和技术,包括实时系统软件设计和用工控组态软件平台进行软件开发。在内容的叙述上,以理论联系实际为原则,注重简明扼要,通俗易懂。由于软件工程是一门实践性很强的学科,书中提供了许多范例供读者参考。

本书共分 16 章。第 1 章软件工程的基本概念,第 2 章需求工程,第 3 章软件测试,第 4 章软件维护与软件进化,第 5 章软件工程管理,第 6 章软件设计基础,第 7 章 UML 建模语言,第 8 章实时软件设计基础,第 9 章组态软件

技术基础,第 10 章工控过程的图形画面设计,第 11 章实时数据库与变量组态,第 12 章输出报表组态,第 13 章通讯组态,第 14 章工控系统中的设备管理,第 15 章工控组态软件的通讯网络,第 16 章工控组态软件设计应用实例。每章后附有习题。

本书由赵一丁主编,聂钢主审。参加编写的有范立娜、赵玉倩、刘秀田、曹毅。本书第 1、8、9、14、15 章由赵一丁编写,第 3、4、5、6 章由范立娜编写,第 11、12、13、16 章由赵玉倩编写,第 2 章由刘秀田编写,第 7 章由曹毅编写。全书由赵一丁统稿,西安交通大学聂钢教授审阅全稿。聂教授对本书提出了许多宝贵意见和建议,对此表示衷心的感谢。在本书的编写过程中,我们还得到了汪晋宽教授和江南大学张秋菊教授的热情关心和指导,在此一起表示感谢。

本书参考教学时数为 56 学时,其中授课 46 学时,上机实习 10 学时。

限于编者水平,书中的缺点和错误在所难免,敬请广大读者批评指正。

编 者

2006 年 7 月

# 第 1 章 软件工程的基本概念

## 1.1 引言

现在,人们工作、生活、生产的各个方面都离不开计算机系统。计算机系统包括计算机硬件和软件两部分。计算机硬件在软件的控制下进行逻辑处理和数学运算,实现规定的功能。初期的软件只是简单的程序,软件的开发即是程序的设计。开发人员的组织形式是个体的。随着硬件规模的不断增长,软件的规模和复杂度也随之增加,软件从早期的程序发展成复杂的软件产品,软件的开发逐步产业化,参与软件开发的人员越来越多,软件开发开始面临许多新的问题,如软件开发的规范化管理、软件的开发方法和技术、软件开发的质量管理、开发成本控制等。软件工程即是为适应软件的产业化发展需要,逐步发展起来的一门有关软件项目开发的工程方法学。

### 1.1.1 什么是软件

计算机软件不仅仅是程序。目前,计算机软件公认的解释是程序、数据及其相关文档的完整集合。也就是说,软件由两部分组成:程序与文档。程序是机器可以执行的程序及有关的使程序能正常操纵信息的数据,文档是用于描述程序和数据以及使用和维护程序所需要的图文资料。

按软件功能可将软件划分为系统软件、支撑工具软件和应用软件。

#### 1. 系统软件

系统软件是计算机系统的底层管理软件,它与计算机硬件紧密配合,管理与硬件相关的数据输入输出和数据处理,使计算机系统的各个部分协调、高效地工作。例如:操作系统、数据库管理系统等。

#### 2. 支撑工具软件

包括用于协助用户软件开发和进行软件测试、维护的工具性软件(如:源程序代码编辑器、程序代码编译器、程序调试器、构件库)和用于工业监控系统软件开发的工控组态软件。

#### 3. 应用软件

根据用户需求为最终用户提供事务处理、工程计算、信息管理、工业控制的软件。如:文字处理软件、CAD 软件、城市交通监管系统、生产设备的自动控制系统软件等。

### 1.1.2 软件危机

在 20 世纪 60~70 年代,出现了软件危机。所谓软件危机,是指在软件开发和维护过程中所遇到的一系列严重问题。随着计算机硬件技术的进步和用户需求愈来愈复杂,一

些复杂的、大型的软件开发项目被提出来了,但软件开发技术的进步一直未能满足发展的要求。在软件开发中遇到的一些问题找不到解决的办法,问题积累起来便形成了尖锐的矛盾,失败的软件开发项目屡见不鲜,从而导致了软件危机。软件危机主要表现在以下几个方面:

(1) 主观盲目地制订软件开发计划,对工作量估计不足,进度计划无法遵循,使得开发工作的完成时间一再拖延,经费预算经常超支。

(2) 不重视软件测试工作,提交给用户的软件质量差,在运行中暴露出大量的问题。在应用领域工作的不可靠软件,轻者影响系统的正常工作,重者发生事故,甚至造成生命财产的重大损失。

(3) 开发过程没有统一的规范化方法和管理流程,设计和实现过程的文档资料不完整,开发人员各行其是,相互之间接口的统一问题常常被忽视,软件结构不清晰。这些都使得软件难以维护。

### 1.1.3 什么是软件工程

软件工程学作为指导软件开发与维护实践的理论,是在人们为解决 20 世纪 60 年代开始出现的软件危机中逐步形成和发展起来的。1968 年,在联邦德国召开的计算机国际会议上,计算机专家们专门针对软件危机问题进行了讨论,正式提出并使用了“软件工程”这一术语。于是,软件工程作为一门新兴的工程学科诞生了。软件工程是一门工程方法学,涉及到从软件需求分析到开发、维护的所有阶段。

当今社会正处在软件产业高速发展的时期。以软件为特征的产品不断涌现,渗透到社会生活、工业、服务业和人们日常生活的各个方面。许多以软件为核心的计算机系统已成为关键性系统,人们对其已形成严重的依赖性。如银行计算机客户的账户管理系统,一旦软件出了问题,可能会使客户的账户资金丢失;再如化工厂的计算机生产过程控制系统,如果出了故障,就可能引起有害气体泄漏或爆炸,造成人身伤害。在这样的系统中,软件的可靠性和可维护性至关重要。

人们寄希望于通过软件工程解决软件危机现象,使开发出的软件产品具有高的可靠性和可维护性,而且要设法降低开发成本,缩短交付周期。但是,软件危机现象并没有得到彻底消除。一些老的危机问题可能解决了,但接着又出现了许多新的危机问题,于是人们不得不去寻找一些更新更好的软件工程方法。正是软件危机问题的不断出现,推动着软件工程方法学的不断发展。

## 1.2 软件工程

### 1.2.1 软件工程的目標

软件工程是一门关于软件开发与维护的工程学科。1983 年,国际权威机构 IEEE 给出了软件工程的定义:软件工程是开发、运行、维护和修改软件的系统方法。其中的软件被定义为:计算机程序、方法、规则、相关的文档资料以及计算机程序运行时所需要的数

据。Fairly 给出的定义是:软件工程学是为了在成本限额以内按时完成开发和修改软件产品所需要的系统生产和维护的技术及管理学科。

软件工程的目的是基于具体软件项目的目标而提出来的。具体地说包括以下几方面:

- (1) 软件功能满足用户的需求。
- (2) 软件性能好。例如软件执行速度要快。
- (3) 软件可靠性高。
- (4) 软件易于操作、使用和维护。
- (5) 降低软件对硬件的依赖性,软件移植性好。
- (5) 降低软件的开发成本。
- (6) 缩短开发周期,使软件及时交付使用。

同时使上述目标达到理想程度往往是难以做到的。在具体的项目实施中,各个目标之间有可能是相互冲突的。如果只顾降低开发成本,可能会导致软件性能和可靠性的下降。如果过分追求软件的可靠性,可能会大大延长软件的测试和开发周期,使开发成本上升,迟迟不能交付使用。如果过于追求软件的性能,则可能造成开发出的软件对硬件有较大的依赖,从而直接影响到软件的可移植性。因此,实施软件项目开发的过程就是力图在以上目标的冲突中取得一定程度的平衡的过程。

## 1.2.2 软件工程技术

软件工程技术是指软件工程所具有的技术要素。作为软件开发与维护的工程方法学,软件工程具有三方面的技术要素,即软件工程方法、软件工具和软件工程过程。

### 1. 软件工程方法

软件工程方法是在软件开发与维护的过程中为了保证产品的高质量、低成本和高效率而采用的技术方法。其方法体现在使用图形或某种特殊语言的方式来表现这些任务中需要建立的软件系统模型,如:数据流模型、软件结构模型、对象模型、组件模型等。主要的软件工程方法有:结构化方法、JSD 方法和面向对象方法。

结构化方法是通过软件功能的划分来实现软件过程的建模。包括结构化功能分析、结构化设计和结构化维护。JSD 方法主要用在软件设计上,也是一种以软件功能为对象的软件设计方法。它以软件中的数据结构为基本依据来进行软件结构与程序算法设计,是对结构化设计方法的有效补充。

面向对象方法是以软件问题域中的对象为依据来构建软件系统开发模型。在人们提出的诸多面向对象方法中,由 Booch 和 Rumbaugh 等提出的面向对象的方法成为主流方法,并被集合称为统一建模语言(UML, Unified Modeling Language)。

不同的方法有不同的适用领域。如面向对象的方法常用于面向用户的交互式系统开发,但并不适用于有严格实时性要求的系统中。

### 2. 软件工具

软件工具是为了软件工程方法的应用而提供的具有自动化特征的软件支撑环境。通常也被称为 CASE 工具。软件工具的覆盖面很广,包括:系统需求分析建模、设计建模、

代码编辑编译和调试、软件测试、项目管理等。

### 3. 软件工程过程

软件工程过程是指按照一定的软件工程方法在软件开发过程中所进行的一系列软件工程活动。实际上,这一系列的活动也就是在软件开发中需要制订的工作步骤,它应该是科学的、合理的,否则将影响软件开发成本、进度与产品质量。

软件工程过程并没有统一的模式。软件产品不同,软件工程过程也会有所不同。但一般来说,软件工程过程至少应包括以下基本活动:

- (1) 软件定义:进行软件规格和使用限制的定义。
- (2) 软件开发:根据软件规格定义制作出软件产品。
- (3) 软件验证:确认软件能够满足用户提出的要求。
- (4) 软件维护:修正软件缺陷,并能根据用户需求变化改进软件。

#### 1.2.3 软件工程管理

作为一个工程项目,项目的管理是重要的。软件项目的规模越大,所需要的管理支持工作量越大。可以说,项目的管理质量决定了软件项目的成败。

软件工程管理是对软件项目的开发管理,是对整个软件生存期的所有活动进行管理。任何工程的质量和成败都与管理的好坏密切相关,软件工程更不例外。尤其是软件产品的特殊性,使得软件工程管理对于保证软件产品的质量、控制开发成本具有极为重要的作用。

软件工程管理涉及到软件工程的许多方面。如软件项目的管理,人员的管理,产品质量管理,成本控制等。

#### 1.2.4 软件工程文化

软件工程不仅仅是解决工程中的技术问题,随着软件质量要求的提高和软件产业的发展,软件工程不得不考虑工程中的文化因素。如软件工程技术人员在软件开发中所应该具备的产品质量意识、价值观、道德准则和团队意识等。

软件工程文化首先体现为工程思想与工程行为。软件工程文化是通过一系列具有工程特征的制度、标准展现出来的,例如:工作制度、工程规则、质量标准等。这些不仅仅是要形成文档和文件,而且需要被所有员工学习、理解,并成为他们的行为指南。当诸多工程思想被所有员工接受而影响到他们的行为,并由此转化为软件产品的质量与价值时,软件工程文化的作用也就体现出来了。

由于软件的开发依赖于人的智力劳动,开发人员的个体行为往往对软件质量的影响很大。软件工程人员必须对自己的行为进行约束,使之在法律和道德准则的框架内。如软件工程人员要正确评估自己的能力,不要承担超出自己能力的开发工作;不编写恶意代码,不传播病毒;不侵犯他人的知识产权,注意保护企业和客户的知识产权;保守企业和客户的技术和商业秘密。

## 1.3 软件开发过程

### 1.3.1 软件开发的特点

同硬件产品一样,软件也是产品,可以在市场上独立于硬件出售。硬件产品的开发过程已高度规范化。首先是硬件产品的总体方案设计、详细的图纸设计和工艺设计,然后是产品的生产,产品经质量检验合格后就可以销售出去。软件的开发也像硬件一样,在软件分析、设计完成之后,编制程序代码,最终形成软件产品。

但是,软件的开发有其自己的特点。软件是通过人们的智力活动,把知识与技术转化成信息的一种产品。与硬件相比,软件的开发更依赖于开发者的业务素质和能力,人员的组织、合作和管理。

软件开发的特点主要体现在以下几个方面:

#### 1. 软件生产的自动化不高

直到今天,软件开发仍然离不开工程人员的个人创造与手工编程操作,软件生产仍不可能像硬件设备的生产那样,达到高度的自动化。同硬件产品的生产相比,生产周期不易控制。

#### 2. 软件生产成本不易控制

由于硬件产品的生产工艺是确定的,生产成本易于核算和控制。但是,软件开发是由软件开发人员开发完成的,所需的开发人员人数和开发周期,软件测试和维护的工作量难于准确估算,使得软件生产成本不易控制。

#### 3. 软件的质量难以保证

软件在开发过程中,尽管经过了严格的测试和试用,但仍然不能保证软件是没有错误的。软件不同于硬件,它是计算机系统中的指令逻辑,是一种逻辑产品,缺乏“可见性”。硬件设备的错误或缺陷往往可以通过它的物理现象直接反映出来,或用测量仪器直接检测出来。但软件错误没有这些直观表现,如软件中存在的某些程序行错误,就必须等到这行程序执行时才有可能被发现。因此,软件错误比起硬件错误来更难发现。所以,软件投入使用后仍需要进行维护,这就带来软件维护复杂性的问题。软件产品维护比硬件产品维护复杂得多。

### 1.3.2 软件开发规格文档

计算机软件不仅仅是程序,在软件开发过程中还应该产生出一系列的文档资料。软件文档是指某种数据媒体和其中所记录的数据,它具有永久性。在软件开发过程中,软件开发机构的管理人员需要使用这些文档资料来管理软件项目;技术人员则需要利用文档资料进行信息交流,提高开发效率。文档记录了开发过程中的有关信息,便于技术人员协调软件开发、软件测试和软件维护;用户通过这些文档资料能够认识软件,对软件进行验收,熟悉软件的安装、操作等。

按照软件文档的产生和使用范围,可以将软件文档分为以下三类:

### 1. 开发生档

软件开发过程中,作为软件开发人员前一阶段工作成果的体现和后一阶段工作依据的文档。包括可行性研究、项目开发计划、需求说明、数据说明、概要设计和详细设计。

### 2. 管理文档

软件开发过程中,由软件开发人员制订的、需提交管理人员的一些工作计划和工作报告,包括项目开发计划、测试计划、测试报告、开发进度月报及项目开发总结。

### 3. 用户文档

软件开发人员为用户准备的有关该软件使用、操作、维护的资料。包括用户手册、操作手册、维护修改建议、需求说明。

按照计算机软件产品开发文件编制指南的国家标准(GB 8567—88)的要求,在计算机软件项目的开发过程中,一般应产生以下文件:可行性研究报告、项目开发计划、需求规格说明书、测试计划、概要设计说明书、数据库设计说明书、详细设计说明书、模块开发文档、用户操作手册、系统维护手册、测试分析报告、系统试运行计划、开发进度月报、项目开发总结报告。

## 1.3.3 软件复用技术

复用又称为重用或再用。在工业工程中,大部分项目都使用了复用技术,其设计过程几乎都是基于构件复用的,即把其他系统使用的经过验证的构件用于新系统。如一个电气控制系统,在设计时会选用标准的电机和电气开关、通用的可编程控制器 PLC、通用的电气柜。通过构件的复用,提高了系统的开发效率和质量,降低了成本。复用技术被用于软件工程,一直是人们关注的热点之一。软件复用是降低软件成本、提高软件生产率和软件质量的有效途径。

软件工程项目的开发由于依赖于人的智力劳动而有其自身的特殊性。在许多的软件项目中,一切工作从头开始。也许这种特有的现象正是软件开发效率低下和质量问题严重的根源所在,正是软件成本高昂的因素之一。因此,开发人员一直企盼这一局面有所改变。

C 语言同机器语言和汇编语言相比,在功能上的复用有了进步。C 语言并不仅仅是语言本身,还包括许多功能函数,可供开发人员调用,使开发人员在编程时不必一切都从头做起,减轻了开发人员的劳动。

C++是在 C 语言基础上发展起来的一种现在被广泛使用的面向对象的编程语言。为什么面向对象程序设计方法现在被人们所接受认可并得到广泛的应用呢?从技术上看,面向对象方法提供的继承性、委托机制、多态性以及类属等机制,使其以类为封装的单位,达到了类级代码的复用。

软件构件技术的出现使软件的复用更进一步。构件是对一组类的组合进行封装,并代表完成一个或多个功能的特定服务,也为用户提供了多个接口。整个构件隐藏了具体的实现,只用接口提供服务。这样,在不同层次上,构件均可以将底层的多个逻辑组合成高层次上的力度更大的新构件,甚至直接封装到一个系统,使模块的重用从代码级、对象级、架构级到系统级都可能实现,从而使软件像硬件一样,能够实现自由装配定制。基于

构件的软件开发技术近年来取得了突飞猛进的发展,对软件产业的技术革新产生了深远的影响。

按功能划分,可将构件分为 3 个层次:底层为基本数据类构件和系统支撑构件,包括数据库构件、代码表构件;中间层为各种通用的构件包括消息中间件、数据交换中间件等;顶层为针对各种领域的专用构件或子系统构件,例如统计构件、查询构件等。

至今,构件技术已形成 3 个流派:Sun 的 Java 平台,Microsoft 的 COM+、OLE、ActiveX, .NET 平台,IBM 的 CORBA 平台。

### 1.3.4 计算机辅助软件工程

在软件开发过程中,软件的开发和管理人员按照软件工程的方法和原则,借助计算机及其软件工具的帮助,开发、测试、维护、管理软件产品的过程,称为计算机辅助软件工程(CASE,Computer-Aided Software Engineering)。采用 CASE 技术和工具的目的是提高软件生产效率和改进软件的质量。

CASE 技术通过使软件开发的某些过程自动化和为开发者提供有用的帮助信息,来为软件开发提供支持。例如:

#### 1. 项目管理

项目管理者利用项目管理工具可以有效地估算软件项目所需的工作量、成本和研制周期等,并制订可行的项目开发计划;根据需求跟踪项目的开发情况;采集度量数据,以此评价软件开发效率和软件产品质量。

#### 2. 系统分析和设计

利用系统分析和设计工具,以图形化的方式建立待开发系统的模型,并评价模型的质量。通过对模型进行一致性和有效性检查,保证分析与设计的完整性。

#### 3. 程序代码的自动生成

利用可视化图形界面设计工具,快捷方便地生成用户图形界面,并产生相应的程序代码;利用代码模板和智能代码感知技术,自动生成部分程序代码。

#### 4. 程序调试

在程序的运行过程中通过设置断点和提供相关数据信息实现对程序的调试。

#### 5. 程序的自动转换

自动完成程序从旧版本编程语言程序到新版本编程语言程序的转换。如从 Visual Basic 6.0 源程序到 Visual Basic .NET 源程序的自动转换。

CASE 技术现在已应用于软件开发过程的各个环节。采用 CASE 工具来完成软件项目从需求分析、设计、实现、测试、维护到管理的全过程,可以大大加速项目开发的速度,并保证软件产品的质量。而集成化 CASE 环境的运用使得开发效率和软件质量的提高更加明显。

但是,运用 CASE 所带来的好处受到以下因素的限制:首先,软件开发本质上是基于人的智力的一种创造性劳动,现有的 CASE 工具尚不能取代人的创造性设计过程;再者,在大型软件项目中,软件工程是团队的合作过程,软件开发人员要花许多时间同团队的其他人员进行交流,CASE 工具的使用在这方面并不能提供更多的帮助。

## 1.4 软件生命周期

生命周期是软件工程的一个重要概念。如同任何事物都有一个发生、发展、成熟直至衰亡的全过程一样,软件系统或软件产品也有一个目标的提出、定义、开发、运行和维护直至被废弃不用的全过程。我国国家标准《计算机软件开发规范》(GB 8566—8)将软件生命周期定义为软件定义、软件开发、软件运行维护 3 个时期,并可以细分为可行性研究、项目计划、需求分析、概要设计、详细设计、编码实现与单元测试、系统集成测试、系统确认验证、系统运行与维护等几个阶段。如图 1-1 所示。

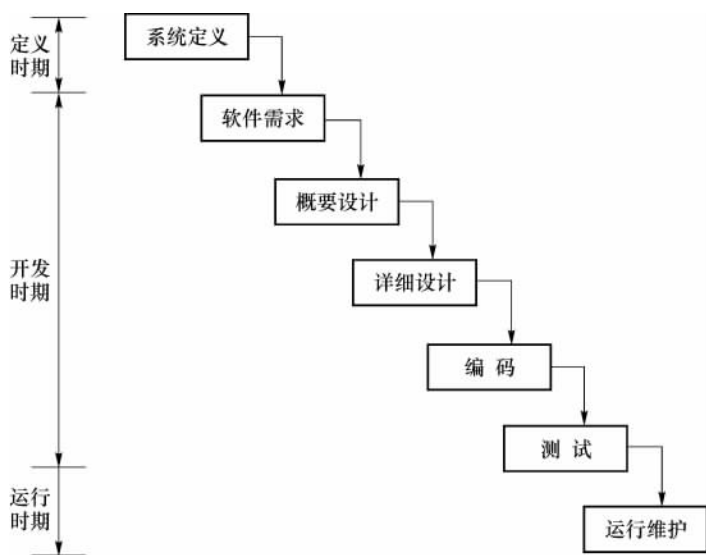


图 1-1 软件生命周期

### 1. 定义时期

这是软件项目的初始时期。其主要任务是分析用户的要求,确定软件开发的总目标,给出系统功能、性能结构、可靠性以及接口等方面的要求。在与用户充分交流的基础上,研究该项软件任务的可行性分析,制订软件项目的开发计划,最终提交项目可行性报告和项目规划报告。

### 2. 开发时期

如同硬件产品的生产过程一样,在系统定义完成之后,就要进行软件产品的设计和生产。

软件的设计任务包括软件功能需求分析、概要设计和详细设计。软件的生产就是软件的编码实现和软件测试验证。

### 3. 运行时期

软件交付给用户使用后就进入运行时期。硬件产品交付使用后由于机械磨损或物理化学性质的改变会出现故障,需要进行维修和预防性维护。软件产品虽然没有磨损现象,

但在运行时期会暴露出在开发阶段遗留的一些问题,所以需要进行软件纠错维护。软件维护还包括适应性维护、功能性维护和预防性维护。适应性维护是指当系统运行环境发生变化以后,为适应这种改变必须对软件进行的修改;功能性维护是指在软件过程中为满足用户需求的变化与扩充对软件所做的修改;预防性维护则是指为改善软件将来的可维护性所做的准备工作。

## 1.5 软件开发模型

软件开发模型(Software Development Model)又称软件生存期模型或软件过程模型(Software Process Model),是软件开发实际过程的抽象与概括,是对软件开发过程各阶段之间关系的描述和表示。软件过程模型是软件工程的重要内容,它为软件工程管理提供框架结构和进度表,为软件开发过程提供原则和方法。软件过程有各种各样的模型,如瀑布模型、进化模型、快速原型模型、螺旋模型、喷泉模型、基于可重用构件的模型等。软件开发模型至今仍在不断发展和改进。

### 1.5.1 瀑布模型

瀑布模型是最早被提出和使用的软件开发模型。由于这个模型描述了软件生命的一些基本过程活动,所以它称为软件生命周期模型。这些活动从一个阶段到另一个阶段依次下降,其工作流程形式上很像瀑布,因此人们把它称为瀑布模型。该模型如图 1-2 所示。

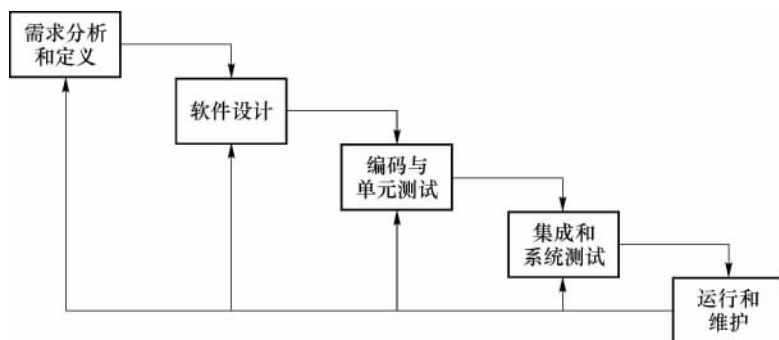


图 1-2 软件生命周期模型

瀑布模型所描述的软件过程活动分为以下阶段：

#### 1. 需求分析和定义

在软件项目进行过程中,需求分析是从软件定义到软件开发的关键步骤,是今后软件开发的基本依据,同时也是用户对软件产品进行验收的基本依据。需求分析和定义是以用户需求为基本依据,从功能、性能、数据、操作等多个方面,对软件系统给出完整、准确、具体的描述,用于确定软件规格。

## 2. 软件设计

根据系统需求的定义,确定系统的结构,进行系统的概要设计和各部分的功能与结构的详细设计。

## 3. 编码与单元测试

在这一阶段,根据软件设计文档完成了程序模块或程序单元的编码。通过程序单元测试,验证其是否满足设计规范。

## 4. 集成和系统测试

程序模块或程序单元被组装集成起来成为一个软件系统,然后进行系统测试。测试完成后即交付用户使用。

## 5. 运行和维护

通常这是软件生命周期中最长的一个阶段。如果在运行期发现了软件的错误,就要修改软件,可能会重复上述某个或多个阶段的活动。

瀑布模型是一种稳定的线性过程。项目被划分为按顺序进行的几个阶段,阶段之间有固定的衔接次序,下一个阶段的开始必须要等到上一个阶段的结束。

瀑布模型的优点之一体现在文档上。在每个阶段都必须完成规定的文档,并在每个阶段结束前都要对所完成的文档进行评审。这种工作方式有利于软件错误的尽早发现和解决,并且对软件系统的后续维护带来很大的便利。

该模型的另一个优点体现在:它所提供的顺序工作流程为软件项目按规程管理提供了便利。如按阶段制订项目计划、分阶段进行成本核算、进行阶段性评审等对提高软件产品质量提供了有效保证。

瀑布模型是一种线性模型,要求项目严格按规程推进,必须等到所有开发工作全部完成以后才能获得可以交付的软件产品,这一过程可能延迟得很长。因此,通过瀑布模型并不能对软件系统进行快速创建,对于一些急于交付的软件系统的开发,瀑布模型并不合适。

瀑布模型主要适合于需求明确且无大的需求变更的软件开发。例如系统软件、实时控制软件。然而,瀑布模型与其他系统工程项目的模型是一致的,当软件项目仅是大型系统工程项目的一部分时,采用瀑布模型是非常合适的。

### 1.5.2 原型进化模型

原型进化模型的思路是:先开发一个原型软件系统交付用户使用,然后根据用户的使用意见和新的需求,对原型系统不断改进,推出新的软件版本,直到推出满足用户需求的最终软件版本,如图 1-3 所示。在这一模型中,软件定义、开发和软件验证是交替进行的,而不是像瀑布模型那样各个阶段是相互独立的。

在没有看到软件的原型之前,用户对软件的需求往往也是模糊不清的。通过对快速开发出的软件原型进行试用,用户对软件的需求会逐渐变得清晰和明确。将这新的需求及时反馈给开发机构,开发人员迅速对软件原型进行改进,及时推出新的软件版本给用户使用。可见,原型进化模型是通过不断发布新的软件版本而使软件逐步完善的。因此,这种开发模式特别适合于那些用户急需的软件产品的开发。它能够快速地向用户交付可以

投入实际运行的软件成果,并能够很好地适应软件用户对需求规格的变更。

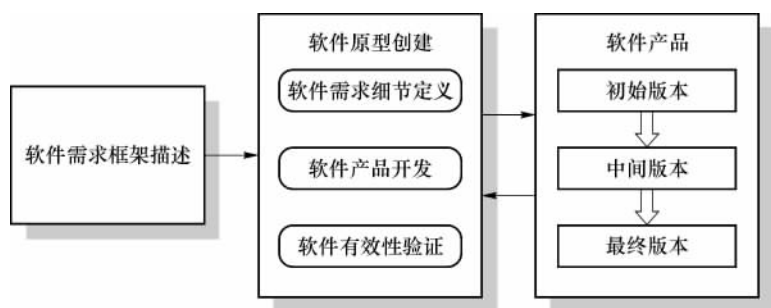


图 1-3 原型进化模型

然而,从工程和管理角度看,原型进化模型存在两方面的问题如下:

(1) 随着开发过程中版本的快速更新,软件文档不能得到及时更改和提交,项目管理者很难掌握软件的开发过程和进度,项目管理、软件配置管理会变得复杂起来。

(2) 由于开发过程中软件版本的不断快速变化,可能损伤软件的内部结构,使其缺乏整体性和稳定性,使软件以后的维护困难。

对于小型和中型软件项目(最多达 500 000 行代码),原型进化模型应该是最好的开发方法。对于大型的、复杂的系统,由于需要不同的开发团队来开发系统的不同部分,而采用原型进化模型要建立一个稳定成熟的系统体系结构是很困难的,这样很难实现开发团队间的协调工作和系统集成。

对于大型系统软件的开发,可采用瀑布模型和原型进化模型的混合开发方法,将上述两种模型的优点结合起来。如用原型进化模型方法快速开发出一个软件原型供用户和开发人员评价,以此来解决软件定义的不确定性问题;对于软件需求明确的系统部分,用瀑布模型来开发;其他部分,如用户界面的设计,事先定义好是有困难的,这时可用原型进化模型进行开发。

### 1.5.3 螺旋模型

螺旋模型是由 Boehn 在 1988 年提出的,如图 1-4 所示。软件开发过程从最内环开始,逐渐向外展开成螺旋型。每一个环代表软件开发的一个阶段。最内环开发出的软件原型用于系统的可行性研究,下一个环路开发出的软件原型主要用于软件的需求定义,接下来的环路进行软件的设计与设计验证等。经过若干的环路的迭代,最终开发出交付用户使用的软件产品。

在螺旋模型中,每一个环路都包括 4 个部分:

#### 1. 目标设定

在这一部分,要确定项目的目标,制订软件定义和详细的项目管理计划,确定项目的风险。

#### 2. 风险评估和规避

对确定的风险进行详细地分析评估,并采取适当的风险规避措施。例如,如果风险是