

高校计算机教学系列教材

软件工程教程

张敬 宋广军 编著
赵硕 王睿

北京航空航天大学出版社

内容简介

软件工程是 20 世纪 60 年代末期为解决“软件危机”而产生的,并迅速发展起来的新兴学科,现在已成为计算机科学技术的一个重要分支。本书主要围绕生命周期法和面向对象方法学,讲述了软件工程的基本概念、基本原理、基本技术、基本方法。由于软件工程是一门实践性很强的学科,书中提供了大量的范例供读者参考。主要内容包括:概述、软件需求分析、软件设计、程序编码、软件测试、面向对象方法学、软件维护、软件标准与软件文档、软件管理、计算机辅助软件工程。内容丰富,图、文示例并茂,通俗易懂。本书可作为高等院校本、专科计算机专业的教材,也可供计算机软件人员和计算机用户阅读。

图书在版编目(CIP)数据

软件工程教程/张敬等编著.—北京:北京航空航天大学出版社,2003.7

ISBN 7-81077-331-3

.软... .张... .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(2003)第 034694 号

软件工程教程

张敬 宋广军 编著
赵硕 王睿

责任编辑 许传安

*

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

北京市松源印刷有限公司印装 各地书店经销

*

开本:787×1092 印张:14.5 字数:371千字

2003年7月第1版 2003年7月第1次印刷 印数:5000册

ISBN 7-81077-331-3 定价:22.00元



总 前 言

科教兴国,教育先行,在全国上下已形成共识。在教育改革过程中,出现了多渠道、多形式、多层次办学的局面。同时,政府逐年加大教育的投入力度。教育发展了,才能有效地提高全民族的文化、科学素质,使我们中华民族屹立于世界民族之林。

计算机科学与技术的发展日新月异,其应用领域迅速扩展,几乎无处不在。社会发展的需求,促使计算机教育生气蓬勃。从普通高校的系统性教学,到远距离的电视、网上教学;从全面讲述,到不同应用领域的、星罗棋布的培训班;从公办的到民办的;从纸介教材到电子教材等等,可以说计算机教学异彩纷呈。要进行教学,就必须有教材。

面对我们这么大的国家和教学形势,在保证国家教学基本要求的前提下,应当提倡教材多样化,才能满足各教学单位的需求,使他们形成各自的办学风格和特色。为此,我们组织北京工业大学、北京航空航天大学、北京理工大学、南开大学、天津工业大学等高校的有丰富教学经验的教师编写了计算机教学系列教材,将陆续与师生见面。

系列教材包括以下各项。

(一) 基础理论:离散数学。

(二) 技术基础:电路基础与模拟电子技术;数字逻辑基础;计算机组成与体系结构;计算机语言(拼盘,选择使用),包括 C++ 程序设计基础、Visual Basic 程序设计基础、Matlab 程序设计基础、Java 程序设计基础、Delphi 语言基础、汇编语言基础等;数据结构;计算机操作系统基础;计算方法基础;微机与接口技术;数据库技术基础等。

(三) 应用基础:计算机控制技术;网络技术;软件工程;多媒体技术等。

(四) 技术基础扩展:编译原理与编译构造;知识工程——网络计算机环境下的知识处理。

(五) 应用基础扩展:计算机辅助设计;单片机实用基础;图像处理基础;传感器与测试技术;计算机外设与接口技术。

本系列教材主要是针对计算机教学编写的,供普通高校、社会民办大学、高等职业学校、业余大学等计算机本科或专科选用。其中一部分教材也适合非计算机专业本科教学使用。在这些教材的内容简介或前言中对使用范围均作了说明。

本系列教材在编写时,注重以下几点:(1) 面对计算机科学与技术动态发展的现实,在内容上应具有前瞻性;(2) 面对学以致用,既有系统的基础知识,又具有应用价值的实用性;(3) 具有科学性、严谨性。另外,力求排版紧凑,使有限的版面具有最大的信息量,以使读者得到实惠。

能否实现这些愿望,只有师生在教学实践中评价。我们期望得到师生的批评和指正。

高校计算机教学系列教材编委会



高校计算机教学系列教材编委会成员

主 任:赵沁平

副 主 任(常务):陈炳和

顾 问:麦中凡

委 员(以姓氏笔划为序):

吕景瑜(北工大教授)

乔少杰(社长,副教授)

麦中凡(北航教授,教育部工科计算机基础教学指导委员会副主任、中专计算机
教学指导委员会顾问)

苏开娜(北工大教授)

陈炳和(北工大教授)

张鸿宾(北工大博导)

郑玉明(北工大副教授)

金茂忠(北航博导)

赵沁平(北航博导,国务院学位办主任)



前 言

随着微电子技术的发展,计算机硬件性能不断提高,价格持续下降。然而长期以来,人们开发优质软件的能力远远落后于硬件技术的发展和用户对计算机软件的需求。自从20世纪60年代末期,为了克服“软件危机”,人们在软件开发领域做了大量的工作,积累了一定的经验,逐渐形成系统的软件开发理论、方法和技术,即软件工程学(通常简称软件工程)。一门新兴学科应运而生,并得到不断的发展。采用工程的方法、技术开发软件,可以提高软件的质量和数量,软件运行期间的维护工作量大大减少了。

进入21世纪信息社会,面对无穷无尽的计算机应用需求,如何更快、更好、更多、更方便地开发软件,已成为软件开发人员面临的主要任务。软件工程的方法和技术越来越受到人们的关注。现在,它已经成为计算机科学技术的一个异常活跃的研究领域,展示了它的勃勃生机。

软件工程是计算机科学的一个重要分支,所涉及的范围非常广泛,包括软件开发技术、软件工程环境、软件经济学、软件心理学和软件工程管理等方面知识。本书以软件的生命周期作为主线索,重点讨论了结构化的软件开发方法和技术,包括结构化分析、结构化设计、编码、测试。在软件工程的入门阶段,结构化软件开发方法是基本、实用的技术。通过对基本概念、基本原理、基本技术、基本方法的学习,使读者能很快运用工程的方法和技术开发软件。近年来,面向对象软件开发方法和技术的研究及应用不断普及,本书利用一定篇幅介绍了面向对象的基本概念、面向对象的分析和设计方法。面向对象方法与人类习惯的思维方式一致,符合人们认识客观世界、解决复杂问题的渐进过程;软件的稳定性、可重用性好及易于维护,是当今比较流行的软件开发方法之一。由于软件工程是一门实践性很强的学科,书中提供了大量的范例供读者参考。书中内容尽量做到通俗易懂,图文并茂,原理、方法与实例相结合。

本书共分10章,第1章:软件工程概述;第2章:软件需求分析;第3章:软件设计;第4章:程序编码;第5章:软件测试;第6章:软件维护;第7章:面向对象方法学;第8章:软件标准与软件文档;第9章:软件管理;第10章:计算机辅助软件工程CASE。其中,第5、6、7章由张敬编写;第1、3章由宋广军编写;第8、9、10章由赵硕编写;第2、4章由王睿编写,全书由张敬统稿。使用本教材的参考讲授学时为30学时,可安排10~20学时组织学生针对具体课题进行设计,以便加



深对软件工程课程内容的理解和掌握。

本书可作为高等院校本、专科计算机专业教材,也可供计算机软件人员和计算机用户阅读。

本书在编写过程中得到了姚春龙博士的大力支持和帮助,在此表示诚挚的感谢。由于时间仓促,加之水平有限,书中难免存在不足之处,恳请广大读者和专家批评指正。

编著者

2002年12月



目 录

第 1 章 软件工程概述.....	1
1.1 软件工程与软件危机	1
1.1.1 软件的发展阶段	1
1.1.2 软件危机	2
1.1.3 软件工程	2
1.2 软件开发模型	3
1.2.1 软件生命周期	3
1.2.2 软件开发的瀑布模型	5
1.2.3 原型化开发模型	8
1.2.4 螺旋模型	9
1.2.5 基于四代技术的模型.....	10
1.2.6 面向对象生存期模型.....	11
1.3 软件计划.....	12
1.3.1 问题定义.....	12
1.3.2 可行性研究.....	13
1.3.3 系统规格说明及评审.....	18
习 题	19
第 2 章 软件需求分析	21
2.1 需求分析概述.....	21
2.1.1 需求分析的任务.....	21
2.1.2 需求分析的步骤.....	22
2.2 需求规格说明书.....	24
2.2.1 数据流图(DFD)	24
2.2.2 数据字典(DD)	26
2.2.3 需求评审.....	29
2.3 结构化分析方法.....	30
2.4 应用示例.....	32
习 题	36
第 3 章 软件设计	38
3.1 软件设计的任务和步骤.....	38
3.2 软件设计概述.....	40
3.2.1 软件设计模块化.....	40



3.2.2	用于软件设计的图形工具.....	43
3.3	结构化系统设计.....	45
3.3.1	软件结构的典型形式.....	45
3.3.2	建立初始结构图.....	46
3.4	详细设计.....	53
3.4.1	详细设计的目的和任务.....	54
3.4.2	结构化程序设计.....	54
3.4.3	详细设计的描述工具.....	57
3.4.4	Jackson 程序设计方法	61
3.4.5	程序结构复杂度的定量度量.....	67
3.5	人机界面设计.....	70
3.5.1	用户的使用需求分析.....	71
3.5.2	人-机界面的设计原则	72
3.5.3	人机界面实现的原则.....	74
3.6	软件设计复审.....	76
习 题	77
第 4 章	程序编码	79
4.1	编码的目的.....	79
4.2	程序设计语言.....	80
4.2.1	程序设计语言分类.....	80
4.2.2	程序设计语言的特征属性.....	82
4.2.3	程序设计语言的准则.....	83
4.3	程序设计风格.....	84
4.3.1	使用好程序内部的文档.....	84
4.3.2	数据说明原则.....	85
4.3.3	语句结构的规则.....	85
4.3.4	输入/输出的规则	86
4.3.5	高效率的原则.....	86
习 题	87
第 5 章	软件的测试	89
5.1	基本概念.....	89
5.1.1	软件测试目标.....	89
5.1.2	软件测试的原则.....	89
5.1.3	软件测试的方法.....	90
5.1.4	软件测试的过程.....	91
5.1.5	软件测试与开发各阶段的关系.....	92
5.2	测试用例的设计.....	93



5.2.1	逻辑覆盖.....	93
5.2.2	等价分类法.....	96
5.2.3	边界值分析法.....	98
5.2.4	错误猜测法.....	99
5.2.5	实用综合测试策略.....	99
5.3	测试步骤.....	102
5.3.1	单元测试.....	102
5.3.2	集成测试.....	104
5.3.3	验收测试.....	107
5.3.4	系统测试.....	108
5.4	调试技术.....	109
5.4.1	调试过程.....	110
5.4.2	调试技术.....	110
5.4.3	调试原则.....	112
	习 题.....	113
第 6 章	软件维护.....	115
6.1	软件维护的种类.....	115
6.1.1	完善性维护.....	115
6.1.2	适应性维护.....	115
6.1.3	纠错性维护.....	115
6.1.4	预防性维护.....	116
6.2	软件维护的特点.....	116
6.2.1	软件工程与软件维护的关系.....	116
6.2.2	维护的成本.....	118
6.2.3	可能存在的问题.....	118
6.3	维护任务的实施.....	119
6.3.1	维护组织.....	119
6.3.2	维护报告.....	119
6.3.3	维护过程.....	120
6.3.4	维护记录的保存.....	121
6.3.5	对维护的评价.....	121
6.4	软件的可维护性.....	122
6.4.1	影响软件可维护性的因素.....	122
6.4.2	提高软件的可维护性方法.....	122
6.5	软件维护的副作用.....	124
6.5.1	修改代码的副作用.....	124
6.5.2	修改数据的副作用.....	125
6.5.3	修改文档的副作用.....	125



习 题.....	126
第 7 章 面向对象方法学.....	127
7.1 面向对象技术的基本概念	127
7.1.1 传统方法学的缺点	127
7.1.2 面向对象技术的由来	128
7.1.3 基本概念	129
7.2 面向对象分析	132
7.2.1 标识类及对象	132
7.2.2 确定结构	134
7.2.3 定义主题	136
7.2.4 标识对象的属性和关联关系	136
7.2.5 标识对象的行为	138
7.2.6 动态模型	140
7.2.7 功能模型	142
7.2.8 三个模型之间的关系	143
7.3 面向对象设计	143
7.3.1 问题域部分的设计	143
7.3.2 人机交互部分的设计	147
7.3.3 任务管理部分的设计	150
7.3.4 数据管理部分的设计	151
习 题.....	153
第 8 章 软件标准与软件文档.....	154
8.1 软件工程标准框架	154
8.1.1 软件工程功能标准框架	154
8.1.2 基于软件生存周期过程的软件工程路径框架	159
8.1.3 我国软件工程标准体系框架	159
8.2 国际软件工程标准	160
8.2.1 IEEE Software Engineering Standards	160
8.2.2 ISO/ IEC/ IEEE/ EIA 12207	162
8.2.3 SEL Recommended Approach	164
8.2.4 SSDM Standards and Procedures	165
8.2.5 ISO 9000 Suite	167
8.2.6 对五种标准的比较	168
8.3 软件文档	171
8.3.1 软件文档	171
8.3.2 文档编制的质量要求	171
习 题.....	173



第 9 章 软件管理.....	174
9.1 软件的目标与项目计划	174
9.1.1 软件项目的特点和软件管理的职能	174
9.1.2 软件目标资源需求的分析和计划	176
9.1.3 软件质量及管理措施	178
9.1.4 项目计划的内容	179
9.2 成本估算	179
9.2.1 自顶向下估算	180
9.2.2 自底向上估算	180
9.2.3 算法模型估计	181
9.3 进度计划	182
9.3.1 Gantt 图(横道图)	182
9.3.2 工程网络	183
9.3.3 估算进度	184
9.3.4 关键路径	184
9.3.5 机动时间	185
9.4 人员分配	185
9.5 软件配置管理	187
9.5.1 软件配置管理任务	187
9.5.2 标识配置对象	187
9.5.3 版本控制	188
9.5.4 系统建立	188
9.5.5 修改控制	188
9.5.6 配置审计	188
9.5.7 配置状况报告	189
9.6 软件质量	189
9.6.1 软件质量的内涵	189
9.6.2 软件质量保证	190
9.6.3 软件质量经济性和质量不断改进	192
9.6.4 软件工程标准及应用	193
9.7 软件工程文件规范	194
习 题.....	198
第 10 章 计算机辅助软件工程 CASE	199
10.1 CASE 概述	199
10.1.1 CASE 的涵义	199
10.1.2 CASE 分类	199
10.2 软件开发工具.....	199



10 2 1	什么是软件工具.....	199
10 2 2	软件开发工具的分类.....	200
10 3	软件工程环境.....	202
10 3 1	软件工程环境的定义.....	202
10 3 2	软件工程环境的分类.....	202
10 3 3	集成化 CASE	203
10 4	大型软件开发环境简介.....	210
10 4 1	JB2 总体结构	211
10 4 2	环境集成机制的主要部件.....	211
10 4 3	工具结构模型和环境中的工具.....	215
	习 题.....	217
	参考文献.....	218



第 1 章

软件工程概述



1.1 软件工程与软件危机

1.1.1 软件的发展阶段

任何一种计算机系统都包括硬件和软件两大部分。硬件只是提供了计算机的可能性,还必须有支持和管理计算机的软件,系统才能实现计算。因此,软件的发展是与硬件的发展相联系的。

自 1946 年世界上第一台计算机问世以来,计算机软件大致经历了三个发展阶段。它们是:程序设计阶段、软件系统阶段、软件工程阶段。

程序设计阶段:在 20 世纪 50 年代至 60 年代,出现了百万次每秒的大型计算机,主要用于科学研究领域。这期间各大公司主要致力于硬件的设计和生,而软件是为某一专门的应用领域设计的。人们认为软件就是程序,主要由设计者本人利用机器语言或汇编语言,运用子程序、过程、函数等技术手段,开发的小型源程序。这类程序大多结构简单,功能单一,工作可靠性差,由设计者自行维护。此阶段的程序设计活动完全是个人程序设计技术的体现。

软件系统阶段:从 20 世纪 60 年代至 70 年代,随着硬件技术的发展,有了微机,人们可以在 PC 机上进行各种信息的处理。为了信息共享和互通消息,还发展了局域网和广域网,计算机的应用面拓宽,用户增加。在这期间,有了专门的软件公司,软件成为产品。软件的功能、规模日益增大,出现了大量的高级语言。软件的开发工作由开发小组所承担,不再是个人的技术满足少数用户的需求。采用的技术手段主要是结构化程序设计。在这一阶段,软件规模的增长,带来了它的复杂度的增加。软件可靠性往往随规模的增长而下降,质量保证也越来越困难,软件的发展速度远不能满足用户的需求,出现了软件危机;而硬件的迅速发展,又要求以“大程序系统”为特征的软件支持。随着计算机应用的增加,尽管提出了软件工程的方法来解决软件的开发和维护问题,但这一问题至今仍然严重地影响着软件的发展。

软件工程阶段:大约在 20 世纪 70 年代以后,以 PC 机为主的计算机已渗透到人类活动的各个领域。这期间出现了大量的兼容机和软件生产企业,除系统软件外,还有工具软件和应用软件,门类繁多,无所不有。许多软件厂商为了使自己的软件能够适应不同的应用,就不断地增加功能,从而使软件规模越来越大,结构越来越复杂。随着软件需求的规模、数量剧增和交付要求的迫切性,大型程序的设计已成为工程项目。其开发工作一般由开发小组及大中型软件开发机构来完成,开发手段也日益丰富,主要包括数据库、开发工具、开发环境、工程化开发方法、标准和规范、网络和分布式开发、对象技术等,并具备了专业维护人员,面向广大市场用



户的需求。虽然开发技术有了很大的进步,但始终未有突破性进展,价格昂贵,并未完全摆脱软件危机。

1.1.2 软件危机

从软件发展的第二个阶段开始,就出现了软件危机。软件的生产不能满足日益增长的软件需求。随着计算机应用的增长,硬件技术迅速发展,出现软件供不应求的局面;更严重的是,软件的生产率随软件规模的增加和复杂性的提高而下降,导致软件的成本在计算机系统成本构成中所占比例急剧上升。庞大的软件费用,加上软件质量的下降,对计算机应用的继续扩大构成巨大的威胁。面对这种严峻的形势,有识之士发出了软件危机的警告。软件危机主要有下述一些表现。

(1) 系统实际功能与实际需求不符。软件开发人员缺乏对用户需求的深入了解,具体实现的功能与用户需求相差太远,导致程序上机运行时出现错误。由于软件人员和用户未能及时交换意见,使得一些问题不能及时解决,而隐蔽下来,造成开发后期矛盾的集中暴露,给将来的调试和维护工作带来更大的困难。

(2) 软件的维护费用急剧上升。软件的费用不仅花费在开发上,尤其要花费在维护上。由于开发阶段存在一定的隐患,因而不能保证软件运行中不再出现错误。维护最重要的事就是纠正软件中遗留的错误,此外,还要进行完善性维护和适应性维护。不言而喻,软件的规模愈大,维护的成本就愈高。

(3) 对软件文档配置没有足够的重视。由于开发过程没有统一的、公认的方法论和规范作指导,软件文档不规范、不健全,参加的人员各行其是,忽视人与人的接口部分。发现了问题修修补补,这样的软件很难维护,提交给用户的软件质量较差。软件文档主要是开发过程各个阶段的说明书、数据字典、程序清单、软件使用、维护手册、软件测试报告及测试用例。这些文档的不完整,是造成软件开发进程、成本不可控、软件维护、管理困难的重要因素。

1.1.3 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。“软件工程”一词,是 1968 年北大西洋公约组织(NATO)在联邦德国召开的一次会议上首次提出的。软件工程是大型软件开发所必须采用的一种重要手段。它采用工程的概念、原理、技术和方法来开发和维护软件,把经过时间考验而证明是正确的管理技术和当前能够得到的最好技术方法结合起来,称之为软件工程。

软件工程是指导计算机软件开发与维护的工程学科。其目标是追求软件产品的正确性、可用性和软件生产的效率。一般情况下,采用生命期方法和结构系统分析、结构系统设计技术来研究软件工程。

利用生命期方法学,从时间角度把软件开发和维护的复杂问题,划分成若干个阶段。每个阶段均有相对独立的任务,然后再逐步完成每个任务,并且每个阶段都要有技术和管理审查,避免反复返工。每个阶段还应有高质量的文档资料,保证软件工程结束后,交付给用户一个完整准确的软件。每一个设计任务还要严格依照结构分析设计技术来完成,以保证软件的质量,



提高软件的可维护性和可靠性。



1.2 软件开发模型

1.2.1 软件生命周期

软件生命周期是一个软件系统从目标提出、定义、开发、使用和维护,直到最后丢弃的整个过程。生命周期是软件工程的一个重要概念。把整个生存期划分为较小的阶段,是实现软件生产工程化的重要步骤。阶段的划分使得人员分工职责清楚,项目进度控制和软件质量得到确认。原则上,前一阶段任务的完成是后一阶段工作的前提和基础;而后一阶段的任务是对于前一阶段问题求解方法的具体化。给每个阶段赋予确定而有限的任务,就能够简化每一步的工作内容,使软件复杂性变得较易控制和管理。

一般来说,软件生命周期包括计划、开发和运行三个阶段。各阶段的划分如图 1.1 所示。

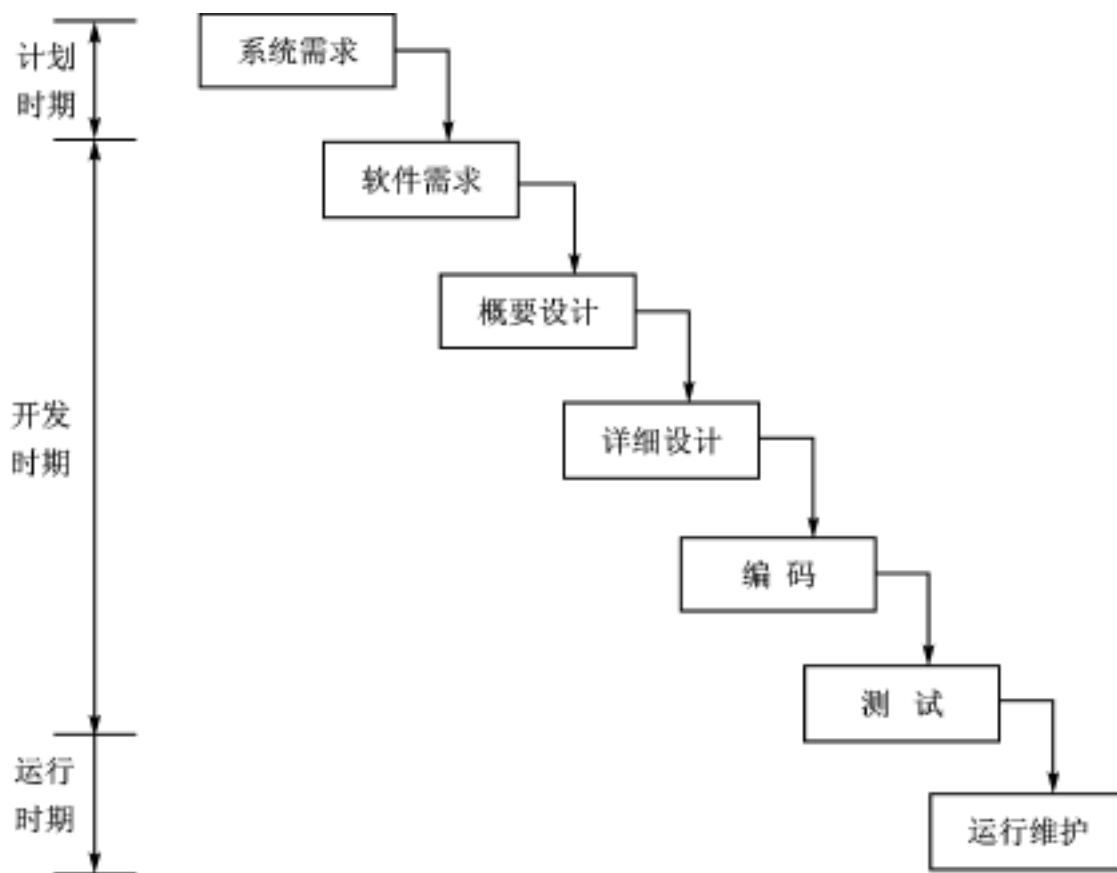


图 1.1 软件生命周期的阶段划分

1. 计划时期

计划时期的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。由分析员和用户合作,研究完成该项软件任务的可行性,制订软件开发计划,并对可利用的资源、成本、可取的效益、开发的进度作出估计,制订出完成开发任务的实施计划,连同可行性研究报告,提交管理部门审查,为软件设计提供依据。因此,软件定义进一步分为问题定义和可行性研究。



(1) 问题定义

这是计划时期的第一步,根据用户或者市场需求,提出软件项目目标和规模,即确定“用户要计算机解决什么问题”。由系统分析员根据对问题的理解,提交关于系统目标与范围的说明。

(2) 可行性研究

问题求解目标一经提出,分析员必须对它进行可行性研究。目的是为前一步提出的问题寻求一种或数种在技术上可行,且在经济上有较高效益的解决方法。为此,系统分析员应在高层次上简化需求分析和概要设计,并写出可行性论证报告。对建议实施的软件项目进行成本效益分析是可行性研究的主要任务之一。

同时,计划时期还应制订出人力、资源及进度计划。

2. 开发时期

开发时期要完成设计和实现两大任务。其中设计任务用需求分析、概要设计和详细设计三个阶段完成;实现任务由编码和测试两个阶段完成。把设计和实现分两步,目的是在开发初期让程序人员集中全力搞好软件的逻辑结构,避免过早地为实现的细节分散精力。

(1) 需求分析

需求分析的任务是完整定义系统必须“做什么?”,并用开发人员与用户均能准确理解的语言表达出来。需求分析是软件开发的基础性工作,必须高度重视,谨慎实施。需求分析文档描述了经过用户确认的系统逻辑模型。它既是软件设计实现的依据,同时也是项目最后验收交付的依据。当采用结构化分析方法,需求规格说明书通常由数据流图、数据字典和加工说明等一整套文档组成。软件工程使用的结构分析设计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准确的系统逻辑模型,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。

(2) 概要设计

主要任务是建立软件的总体结构,包括系统功能设计和系统结构设计。

系统功能设计的任务是确定系统其外部规格与内部规格。所谓外部规格包括:系统运行环境、用户可见功能、性能一览表;系统输入/输出数据格式。内部规格是指各主要处理的基本策略;系统文档种类与规格;系统测试总体方案。系统结构设计任务是确定系统模块结构,确立各模块功能划分和接口规范、调用关系,确定主要模块算法和主要数据结构。所以,系统设计员应选择有经验的高级程序员担任,或直接由系统分析员兼任。总体设计说明书中系统功能设计可以采用表格形式给出;而系统结构设计通常由软件结构图或者高层 IPO 图给出。

(3) 详细设计

详细设计是针对单个模块的设计。目的是确定模块内部的过程结构,详细说明实现该模块功能的算法和数据结构,所以有时也称为算法设计。详细设计由高级程序员和程序员担任,按照系统结构将各模块分解到人。详细设计的完成标志是用图形或者伪码描述的模块设计说明书。



(4) 编 码

编码的任务是根据模块设计说明书,用指定的程序设计语言把模块的过程性描述翻译成源程序。与“需求分析”或“设计”相比,“编码”要简单得多,所以通常由编码员或初级程序员担任。系统编码的完成标志是可运行代码和完整的模块内部文档(源程序清单)。前面产生的都属于软件的文档。

(5) 测 试

测试是开发时期的最后一个阶段。其任务是通过各种类型的测试使软件达到预期的要求。按照不同的层次,又可细分为单元测试、综合测试、确认测试和系统测试等步骤。测试是保证软件质量的重要手段。大型软件的测试通常由独立的部门和人员进行。通过对测试结果的分析,要求建立系统可靠性模型,对系统可以达到的各项功能、性能指标进行量化确认。因此,测试阶段的文档称为测试报告,包括测试计划、测试用例与测试结果等内容。

3. 运行时期

运行时期是软件生命周期的最后一个时期。其主要工作是做好软件维护。维护的目的是使软件在整个生命周期内保证能满足用户的需求和延长软件的使用寿命。软件维护的具体活动包括纠错维护、适应性维护、功能性维护和预防性维护。所谓纠错性维护就是改正在软件运行过程中暴露出来的系统遗留的各种错误。这种维护活动在系统交付初期比较频繁,但当系统进入稳定运行期后应该很少发生。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。软件运行稳定以后,维护的主要活动应该是适应性和功能性维护。

虽然没有把维护阶段进一步划分成更小的阶段,但是实际上每一项维护活动都应该经过提出维护要求、分析维护要求、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软件定义和开发的全过程。

1.2.2 软件开发的瀑布模型

瀑布(waterfall)模型也称软件生存周期模型,由 W. Royce 于 1970 年首先提出。根据软件生存周期各个阶段的任务,瀑布模型从可行性研究(或称系统分析),逐步进行阶段性变换,直至通过确认测试并得到用户确认的软件产品为止。瀑布模型上一阶段的变换结果是下一阶段变换的输入,相邻两个阶段具有因果关系,紧密相连。一个阶段工作的失误将蔓延到以后的各个阶段。为了保障软件开发的正确性,每个阶段任务完成后,都必须对它的阶段性产品进行评审,确认之后再转入下一阶段的工作。评审过程发现错误和疏漏后,应该反馈到前面的有关阶段,修正错误,弥补疏漏,然后再重复前面的工作,直至某一阶段通过评审后再进入下一阶段。这种形式的瀑布模型是带有反馈的瀑布模型。

严格按照软件生命周期的阶段划分,顺序执行各阶段构成软件开发的瀑布型模型,是传统的软件工程生存期模式,如图 1.2 所示。