



教育部职业教育与成人教育司推荐教材  
(五年制) 高等职业教育电子信息类教学用书

21世纪高职高专系列规划教材

# 软件工程与项目管理

刘竹林 白振林 卢润彩



北京师范大学出版社

教育部职业教育与成人教育司推荐教材  
21世纪高职高专系列规划教材

# 软件工程与项目管理

刘竹林 白振林 卢润彩

 北京师范大学出版社

## 内容简介

本书的主要内容:第1章主要介绍了软件工程和项目管理的概念,合格的项目管理和合格的项目管理的定义;第2章介绍了项目计划的制定和可行性研究的方法等内容;第3章介绍了需求分析的目标、原则和方法,需求分析的过程,最后给出了如何作好软件需求;第4章介绍了概要设计的原则和设计方法;第5章介绍了详细设计的原则和设计方法;第6章介绍了结构化程序设计方法,程序设计的算法和效率以及软件项目的编码管理和冗余程序设计;第7章介绍了软件项目评审的内容、方法等,给出了软件开发过程的各阶段的评审要求;第8章介绍了软件测试的基本概念、软件测试的组织和软件测试的策略,最后介绍了常用的软件测试工具;第9章介绍了软件质量保证与软件配置管理;第10章介绍了软件项目的文档管理;第11章给出了CMM软件成熟度模型基础简介;第12章给出了PMI项目管理體系简介。

本书供高职类计算机软件及信息管理专业的学生和教师教学使用,也可作为IT业的管理者、软件开发人员、项目管理人员、技术负责人以及研究所工程技术人员的参考书。

## 图书在版编目(CIP)数据

软件工程与项目管理/刘竹林,白振林,卢润彩编著. —北京:北京师范大学出版社,2005.8  
(21世纪高职高专系列规划教材)  
ISBN 7-303-07662-X

I. 软… II. ①刘…②白…③卢… III. ①软件工程-高等学校:技术学校-教材②软件开发-项目管理-高等学校:技术学校-教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2005)第 086358 号

北京师范大学出版社出版发行  
(北京新街口外大街 19 号 邮政编码:100875)

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

出版人:赖德胜

北京京师印务有限公司印刷 全国新华书店经销  
开本:185 mm×260 mm 印张:20.25 字数:390 千字  
2005 年 9 月第 1 版 2005 年 9 月第 1 次印刷  
印数:1~3 000 册 定价:28.00 元

## 出版说明

随着我国经济建设的发展,社会对技术型应用人才的需求日趋紧迫,这也促进了我国职业教育的迅猛发展,我国职业教育已经进入了平稳、持续、有序地发展阶段。为了适应社会对技术型应用人才的需求和职业教育的发展,教育部对职业教育进行了卓有成效的改革,职业教育与成人教育司、高等教育司分别颁布了调整后的中等职业教育、高等职业教育专业设置目录,为职业学校专业设置提供了依据。教育部连同其他五部委共同确定数控技术应用、计算机应用与软件技术、汽车运用与维修、护理等四个专业领域为紧缺人才培养专业,选择了上千家高职、中职学校和企业作为示范培养单位,拨出专款进行扶持,力争培养一批具有较高实践能力的紧缺人才。

职业教育的快速发展,也为职业教材的出版发行迎来了新的春天和新的挑战。教材出版发行为职业教育的发展服务,必须体现新的理念、新的要求,进行必要的改革。为此,在教育部高等教育司、职业教育与成人教育司、北京师范大学等的大力支持下,北京师范大学出版社在全国范围内筹建了“全国职业教育教材改革与出版领导小组”,集全国各地上百位专家、教授于一体,对中等职业、高等职业文化基础课、专业基础课、专业课教材的改革与出版工作进行深入地研究与指导。2004年8月,“全国职业教育教材改革与出版领导小组”召开了“全国有特色高职教材改革研讨会”,来自全国20多个省、市、区的近百位高职院校的院校长、系主任、教研室主任和一线骨干教师参加了此次会议。围绕如何编写出版好适应新形势发展的高等职业教育教材,与会代表进行了热烈的研讨,为新一轮教材的出版献计献策。这次会议共组织高职教材50余种,包括文化基础课、电工电子、数控、计算机教材。其特点如下:

1. 紧紧围绕教育改革,适应新的教学要求。教育部等六部委联合发文确定紧缺型人才培养战略,并明确提出了高等职业教育将从3年制逐渐向2年制过渡。过渡时期具有新的教学要求,这批教材是在教育部的指导下,针对过渡时期教学的特点,以2年制为基础,兼顾3年制,以“实用、够用”为度,淡化理论,注重实践,消减过时、用不上的知识,内容体系更趋合理。

2. 教材配套齐全。将逐步完善各类专业课、专业基础课、文化基础课教

材,所出版的教材都配有电子教案,部分教材配有电子课件和实验、习题指导。

3.教材编写力求语言通俗简练,讲解深入浅出,使学生在理解的基础上学习,不囫囵吞枣,死记硬背。

4.教材配有大量的例题、习题、实训,通过例题讲解、习题练习、实验实训,加强学生对理论的理解以及动手能力的培养。

5.反映行业新的发展,教材编写注重吸收新知识、新技术、新工艺。

北京师范大学出版社是教育部职业教育教材出版基地之一,有着近20年的职业教育教材出版历史,具有丰富的编辑出版经验。这批高职教材是针对2/3年制编写的,同时也向教育部申报了“2004—2007年职业教育教材开发编写规划”,部分教材通过教育部审核,被列入职业教育与成人教育司5年制高职推荐教材。我们还将开发电子信息类的通信、机电、电气、计算机等其他专业,以及工商管理、财会等方面教材,希望广大师生积极选用。

教材建设是一项任重道远的工作,需要教师、专家、学校、出版社、教育行政部门的共同努力才能逐步获得发展。我们衷心希望更多的学校、更多的专家加入到我们的教材改革出版工作中来,北京师范大学出版社职业与成人教育事业部全体人员也将备加努力,为职业教育的改革与发展服务。

全国职业教育教材改革与出版领导小组  
北京师范大学出版社

## 参加教材编写的单位名单

(排名不分先后)

|               |              |
|---------------|--------------|
| 沈阳工程学院        | 常州轻工职业技术学院   |
| 山东劳动职业技术学院    | 河北工业职业技术学院   |
| 济宁职业技术学院      | 太原理工大学轻纺学院   |
| 辽宁省交通高等专科学校   | 浙江交通职业技术学院   |
| 浙江机电职业技术学院    | 保定职业技术学院     |
| 杭州职业技术学院      | 绵阳职业技术学院     |
| 西安科技大学电子信息学院  | 北岳职业技术学院     |
| 西安科技大学机械学院    | 天津职业大学       |
| 天津渤海职业技术学院    | 北京轻工职工职业技术学院 |
| 天津渤海集团公司教育中心  | 石家庄信息工程职业学院  |
| 连云港职业技术学院     | 襄樊职业技术学院     |
| 景德镇高等专科学校     | 九江职业技术学院     |
| 徐州工业职业技术学院    | 青岛远洋船员学院     |
| 广州大学科技贸易技术学院  | 无锡科技职业学院     |
| 江西信息应用职业技术学院  | 广东白云职业技术学院   |
| 浙江商业职业技术学院    | 三峡大学职业技术学院   |
| 内蒙古电子信息职业技术学院 | 西安欧亚学院实验中心   |
| 济源职业技术学院      | 天津机电职业技术学院   |
| 河南科技学院        | 漯河职业技术学院     |
| 苏州经贸职业技术学院    | 济南市高级技工学校    |
| 浙江工商职业技术学院    | 沈阳职业技术学院     |
| 温州大学          | 江西新余高等专科学校   |
| 四川工商职业技术学院    | 赣南师范学院       |

# 前 言

随着信息技术的飞速发展,软件产品的规模也越来越庞大,个人单打独斗的作坊式开发方式已经越来越不适应发展的需要。

软件工程、软件测试、项目管理、软件质量保证这些概念已经越来越被人们重视。在当前国家重点扶持软件产业的发展形式下,笔者从 20 多年的计算机软件工作中不断收集计算机软件开发经验,致力于在我国计算机技术开发标准化方面抛砖引玉,让我国软件产业尽快成熟起来。

项目管理是古老的也是新兴的知识体系。

我们人类的祖先在远古就创建了无数著名项目:金字塔、比萨斜塔、长城和大运河等。现在,我国在项目管理方面已经训练出许多大型项目管理人才,他们分别在三峡工程、南水北调、西气东输、北京奥运、葛洲坝等大型项目中发挥了巨大作用。

各软件企业都在积极将软件项目管理引入开发活动中,对开发实行有效的管理。从概念上讲,软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成,而对成本、人员、进度、质量和风险等进行分析和管理的活动。实际上,软件项目管理的意义不仅仅如此,进行软件项目管理有利于将开发人员的个人开发能力转化成企业的开发能力,企业的软件开发能力越高,表明这个企业的软件生产越趋向于成熟,企业越能够稳定发展,减小开发风险。

刘竹林同志编写了本书的第 2 章、第 7 章、第 8 章、第 9 章、第 10 章、第 11 章、第 12 章,白振林同志编写了第 1 章和第 3 章,卢润彩同志编写了第 4 章、第 5 章和第 6 章。刘竹林对全书进行了统编。

另外,张彩霞同志在本书编写过程中给予了大力支持和帮助,在此表示由衷地感谢!并感谢在编写过程中支持我们的同志们。

最后,感谢广州无线电集团金融电子设备有限公司的领导和同仁,他们提供了本书的软件开发规范编写文档。

编 者  
2005 年 3 月



## 目 录

## 前言

## 第1章 概述 ..... (1)

## 1.1 软件工程概述 ..... (1)

## 1.1.1 软件工程的定义 ..... (3)

## 1.1.2 软件工程的范畴 ..... (3)

## 1.1.3 软件开发的几个模型 ..... (6)

## 1.1.4 软件生命周期 ..... (10)

## 1.1.5 软件工程的目标 ..... (12)

## 1.2 项目管理概述 ..... (13)

## 1.2.1 项目管理的术语 ..... (13)

## 1.2.2 软件项目管理的特点 ... (14)

1.2.3 软件工程项目管理的目标  
..... (15)1.3 项目经理与项目管理  
..... (16)

## 1.3.1 合格的项目经理 ..... (16)

## 1.3.2 合格的项目管理 ..... (18)

## 1.3.3 影响项目成功的因素 ... (20)

## 本章小结 ..... (23)

## 习题与思考题 ..... (23)

## 第2章 项目启动管理 ..... (24)

## 2.1 项目工作计划 ..... (25)

## 2.2 人力资源计划 ..... (26)

## 2.2.1 项目管理的组织模式 ... (26)

## 2.2.2 团队建立的重要内容 ... (27)

## 2.3 硬件资源规划 ..... (27)

## 2.4 软件资源规划 ..... (28)

## 2.5 可行性研究 ..... (28)

## 2.5.1 可行性研究的内容 ..... (29)

## 2.5.2 可行性研究的步骤 ..... (29)

## 2.5.3 项目效益评价 ..... (30)

## 2.6 项目启动管理 ..... (32)

## 2.6.1 项目立项 ..... (32)

## 2.6.2 项目组织的成立 ..... (33)

## 2.7 实训案例 ..... (34)

## 本章小结 ..... (39)

## 习题与思考题 ..... (39)

## 第3章 软件需求分析管理 ..... (43)

## 3.1 软件需求分析概述 ..... (43)

## 3.1.1 软件需求分析的目标 ... (43)

## 3.1.2 软件需求分析的原则 ... (44)

## 3.1.3 软件需求分析的过程 ... (46)

## 3.2 结构化分析方法 ..... (49)

## 3.2.1 结构化分析方法概述 ... (49)

## 3.2.2 数据流图 ..... (50)

## 3.2.3 数据字典 ..... (54)

## 3.2.4 处理说明 ..... (57)

## 3.3 软件快速原型化开发

## 方法 ..... (62)

## 3.3.1 软件原型化方法概述 ... (62)

## 3.3.2 快速原型开发模型 ..... (63)

## 3.4 如何做好需求分析 ..... (65)

3.4.1 需求分析的五个阶段  
..... (65)

## 3.4.2 如何做好需求分析 ..... (66)

## 3.5 实训案例 ..... (70)

## 本章小结 ..... (76)

## 习题与思考题 ..... (77)





|                    |       |
|--------------------|-------|
| <b>第4章 概要设计</b>    | (79)  |
| 4.1 概要设计的任务和过程     | (79)  |
| 4.1.1 概要设计的任务      | (79)  |
| 4.1.2 概要设计的过程      | (80)  |
| 4.2 概要设计原则         | (81)  |
| 4.2.1 模块化          | (81)  |
| 4.2.2 自顶向下逐步求精     | (82)  |
| 4.2.3 模块的高内聚性和低耦合性 | (82)  |
| 4.3 软件结构的设计方法      | (86)  |
| 4.3.1 IPO图         | (86)  |
| 4.3.2 模块结构图        | (88)  |
| 4.3.3 结构化设计方法      | (92)  |
| 4.4 实训案例           | (96)  |
| 本章小结               | (99)  |
| 习题与思考题             | (99)  |
| <b>第5章 详细设计</b>    | (101) |
| 5.1 详细设计的目标和原则     | (101) |
| 5.1.1 详细设计的任务和目标   | (101) |
| 5.1.2 详细设计的原则      | (101) |
| 5.2 详细设计的内容        | (102) |
| 5.2.1 代码设计原则       | (102) |
| 5.2.2 数据库设计        | (103) |
| 5.2.3 界面设计         | (109) |
| 5.2.4 网络结构设计       | (112) |
| 5.3 程序流程图设计        | (114) |
| 5.3.1 N-S图         | (116) |
| 5.3.2 PAD          | (118) |
| 5.3.3 PDL(结构化语言)   | (119) |
| 5.4 实训案例           | (120) |
| 本章小结               | (127) |
| 习题与思考题             | (127) |

|                    |       |
|--------------------|-------|
| <b>第6章 软件编程</b>    | (129) |
| 6.1 结构化程序设计方法      | (129) |
| 6.1.1 结构化程序设计的原则   | (129) |
| 6.1.2 逐步求精设计法      | (130) |
| 6.1.3 结构化程序设计风格    | (130) |
| 6.2 程序设计算法与效率      | (132) |
| 6.2.1 程序设计算法       | (132) |
| 6.2.2 程序的运行效率      | (134) |
| 6.3 程序设计语言的特点与应用范围 | (134) |
| 6.3.1 程序设计语言的发展与分类 | (134) |
| 6.3.2 程序设计语言的选择    | (137) |
| 6.4 冗余程序设计与防错程序设计  | (138) |
| 6.4.1 冗余程序设计       | (138) |
| 6.4.2 防错程序设计       | (138) |
| 6.5 软件编码管理         | (139) |
| 6.5.1 软件编码过程的管理    | (139) |
| 6.5.2 编码后的管理       | (140) |
| 6.6 案例:C/C++代码规范   | (140) |
| 6.6.1 文件结构         | (141) |
| 6.6.2 程序版式         | (142) |
| 6.6.3 命名规则         | (147) |
| 6.6.4 表达式和基本语句     | (149) |
| 6.6.5 常量           | (152) |
| 6.6.6 函数设计         | (152) |
| 6.6.7 重载和内联        | (155) |
| 6.6.8 内存管理         | (155) |
| 6.6.9 类的构造函数、析构函数、 |       |

|                         |              |                       |              |
|-------------------------|--------------|-----------------------|--------------|
| 成员函数与赋值函数 .....         | (156)        | 8.3.1 测试组织者 .....     | (179)        |
| 6.6.10 类的继承和组合 .....    | (156)        | 8.3.2 专业测试人员 .....    | (179)        |
| 6.6.11 其他规范及建议 .....    | (157)        | 8.3.3 测试配置管理人员 .....  | (179)        |
| 本章小结 .....              | (158)        | 8.3.4 测试相关人员 .....    | (179)        |
| 习题与思考题 .....            | (158)        | 8.4 软件测试技术及其用例        |              |
| <b>第7章 软件项目评审 .....</b> | <b>(159)</b> | 设计 .....              | (180)        |
| 7.1 项目评审概论 .....        | (159)        | 8.4.1 软件测试的分类 .....   | (180)        |
| 7.1.1 评审的目的 .....       | (159)        | 8.4.2 白盒子测试及其用例设计     |              |
| 7.1.2 评审人员及其职责 .....    | (160)        | .....                 | (181)        |
| 7.1.3 评审的基本要求和评审        |              | 8.4.3 黑盒子测试及其用例设计     |              |
| 依据 .....                | (160)        | .....                 | (184)        |
| 7.1.4 评审内容 .....        | (160)        | 8.4.4 静态测试 .....      | (188)        |
| 7.1.5 评审方式 .....        | (160)        | 8.4.5 动态测试 .....      | (189)        |
| 7.1.6 评审工作程序 .....      | (161)        | 8.5 软件测试策略 .....      | (189)        |
| 7.2 各阶段的评审内容和           |              | 8.5.1 单元测试 .....      | (190)        |
| 要点 .....                | (164)        | 8.5.2 集成测试 .....      | (191)        |
| 7.2.1 需求分析的评审 .....     | (165)        | 8.5.3 确认测试 .....      | (192)        |
| 7.2.2 概要设计的评审 .....     | (166)        | 8.5.4 系统测试 .....      | (193)        |
| 7.2.3 详细设计的评审 .....     | (167)        | 8.5.5 验收测试 .....      | (194)        |
| 7.3 实训案例 .....          | (169)        | 8.6 软件测试工具 .....      | (195)        |
| 本章小结 .....              | (170)        | 8.7 实训案例 .....        | (196)        |
| 习题与思考题 .....            | (171)        | 本章小结 .....            | (199)        |
| <b>第8章 软件测试技术 .....</b> | <b>(172)</b> | 习题与思考题 .....          | (201)        |
| 8.1 软件测试的基本概念           |              | <b>第9章 软件质量保证与软件配</b> |              |
| .....                   | (172)        | <b>置管理 .....</b>      | <b>(203)</b> |
| 8.1.1 软件测试的重要性 .....    | (172)        | 9.1 软件质量与 SQA .....   | (203)        |
| 8.1.2 软件测试的定义 .....     | (174)        | 9.1.1 SQA 的目标 .....   | (203)        |
| 8.1.3 软件测试与软件调试的        |              | 9.1.2 质量保证计划 .....    | (204)        |
| 区别 .....                | (174)        | 9.2 软件配置管理(SCM)       |              |
| 8.1.4 软件测试的基本原则         |              | .....                 | (206)        |
| .....                   | (175)        | 9.2.1 软件配置的重要性 .....  | (207)        |
| 8.1.5 软件测试的目标 .....     | (177)        | 9.2.2 基线管理 .....      | (208)        |
| 8.1.6 软件测试过程 .....      | (177)        | 9.2.3 软件配置项 .....     | (208)        |
| 8.2 软件测试与软件开发           |              | 9.2.4 软件配置方法 .....    | (209)        |
| 的关系 .....               | (178)        | 9.3 案例——某公司软件         |              |
| 8.3 测试团队的组织 .....       | (179)        | 配置计划 .....            | (211)        |



|                                     |       |                                   |       |
|-------------------------------------|-------|-----------------------------------|-------|
| 9.3.1 介绍 .....                      | (211) | 11.3 CMM 的模型框架 .....              | (264) |
| 9.3.2 组织 .....                      | (211) | 11.4 CMM 的关键过程域 .....             | (264) |
| 9.3.3 配置项与基线 .....                  | (212) | 11.4.1 等级 2 中的关键过程域 .....         | (265) |
| 9.3.4 变更控制 .....                    | (212) | 11.4.2 等级 3 中的关键过程域 .....         | (265) |
| 9.3.5 状态统计 .....                    | (213) | 11.4.3 等级 4 中的关键过程域 .....         | (266) |
| 9.3.6 配置审计 .....                    | (213) | 11.4.4 等级 5 中的关键过程域 .....         | (266) |
| 9.3.7 估计时间表 .....                   | (214) | 11.5 CMM 在国内外的应用 .....            | (266) |
| 9.3.8 资源 .....                      | (214) | 11.5.1 CMM 在美国的应用 .....           | (267) |
| 本章小结 .....                          | (215) | 11.5.2 CMM 在印度的应用 .....           | (267) |
| 习题与思考题 .....                        | (215) | 11.5.3 CMM 在中国的应用 .....           | (268) |
| <b>第 10 章 软件文档的管理</b> .....         | (217) | 11.6 中国软件企业 CMM 的发展 .....         | (269) |
| 10.1 软件文档的作用 .....                  | (217) | 11.7 CMM、ISO9000 与项目管理之间的关系 ..... | (271) |
| 10.1.1 什么是软件文档 .....                | (217) | 本章小结 .....                        | (272) |
| 10.1.2 软件文档的作用 .....                | (217) | 习题与思考题 .....                      | (273) |
| 10.2 软件文档的分类 .....                  | (218) | <b>第 12 章 PMI 项目管理体系简介</b> .....  | (274) |
| 10.2.1 开发文档 .....                   | (218) | 12.1 项目总体管理 .....                 | (275) |
| 10.2.2 管理文档 .....                   | (218) | 12.1.1 项目计划的制定 .....              | (275) |
| 10.2.3 用户文档 .....                   | (218) | 12.1.2 项目计划的实施 .....              | (276) |
| 10.3 软件文档格式 .....                   | (219) | 12.1.3 综合变更控制 .....               | (276) |
| 10.3.1 软件开发类文档格式 .....              | (219) | 12.2 项目工作范围的管理 .....              | (276) |
| 10.3.2 项目管理类文档格式 .....              | (220) | 12.2.1 项目范围计划的编制 .....            | (277) |
| 10.3.3 软件用户文档类格式 .....              | (220) |                                   |       |
| 本章小结 .....                          | (259) |                                   |       |
| 习题与思考题 .....                        | (259) |                                   |       |
| <b>第 11 章 CMM 软件成熟度模型基础简介</b> ..... | (260) |                                   |       |
| 11.1 CMM 简介 .....                   | (260) |                                   |       |
| 11.2 CMM 的五个等级 .....                | (261) |                                   |       |
| 11.2.1 初始级 .....                    | (261) |                                   |       |
| 11.2.2 可重复级 .....                   | (263) |                                   |       |
| 11.2.3 已定义级 .....                   | (263) |                                   |       |
| 11.2.4 定量管理级 .....                  | (263) |                                   |       |
| 11.2.5 优化级 .....                    | (264) |                                   |       |

|                               |       |                      |       |
|-------------------------------|-------|----------------------|-------|
| 12.2.2 项目范围的定义 .....          | (277) | 12.5.3 项目质量控制 .....  | (283) |
| 12.2.3 范围变更的控制 .....          | (279) | 12.6 项目沟通管理 .....    | (284) |
| 12.3 项目时间管理 .....             | (279) | 12.7 项目人力资源的管理 ..... | (284) |
| 12.3.1 项目活动的排序 .....          | (279) | 12.8 项目风险管理 .....    | (288) |
| 12.3.2 项目工期的估算 .....          | (281) | 12.8.1 风险的分类 .....   | (288) |
| 12.3.3 项目进度计划的制定和<br>控制 ..... | (282) | 12.8.2 项目风险的识别 ..... | (289) |
| 12.4 项目成本管理 .....             | (282) | 12.8.3 风险的分析 .....   | (291) |
| 12.4.1 项目资源计划编制 ...           | (282) | 12.8.4 风险的应对措施 ..... | (293) |
| 12.4.2 项目成本预算 .....           | (282) | 12.9 项目采购管理 .....    | (294) |
| 12.4.3 项目成本控制 .....           | (283) | 12.10 实训案例 .....     | (296) |
| 12.5 项目质量管理 .....             | (283) | 本章小结 .....           | (299) |
| 12.5.1 项目质量计划的编制<br>.....     | (283) | 习题与思考题 .....         | (299) |
| 12.5.2 项目质量保证 .....           | (283) | 部分思考题与习题答案 .....     | (300) |

## 第1章 概述

### 本章要点

1. 软件工程概述。
2. 项目管理的基本概念。
3. 软件项目管理的特点。
4. 软件工程项目管理的目标。
5. 合格的项目经理。
6. 合格的项目管理。
7. 影响项目成功的因素。

一般来讲,开发一个项目需要项目管理技术,开发一个软件需要软件工程技术。从计算机软件的发展历程,我们可以看出,项目管理和软件工程是开发软件项目不可分割的并行的两个过程,其中项目管理是项目的管理过程,软件工程是指项目的技术过程。经历了如下几个阶段,如表 1.1 所示。

表 1.1 软件的发展阶段

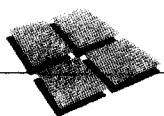
| 发展年代               | 软件的发展阶段 | 软件的含义     | 决定质量的因素     |
|--------------------|---------|-----------|-------------|
| 20 世纪 50 年代至 60 年代 | 程序设计阶段  | 程序及说明书    | 个人技术水平      |
| 20 世纪 60 年代至 70 年代 | 程序系统阶段  | 程序,数据,说明书 | 小组技术水平      |
| 20 世纪 70 年代以后      | 软件工程阶段  | 程序、文档、数据  | 项目经理的项目管理水平 |

### 1.1 软件工程概述

自 1946 年第一台计算机问世以来,计算机学科就成为当今世界发展最快新兴学科,在近 60 年的时间里,计算机已经渗透到社会的各个领域,有力地推动了整个社会信息化的发展。

进入 21 世纪以后,中国的高技术产业,特别是信息产业的持续高速发展,使信息技术已经成为新世纪经济发展的驱动力。软件是信息化的核心,国民经济、国防建设、社会发展及人民生活都离不开软件。软件产业是增长最快的朝阳产业,是高投入/高产出、无污染和低能耗的绿色产业。软件产业关系到国家经济和文化安全,体现了国家综合实力,是决定 21 世纪国际竞争地位的战略性产业。

软件工程是一门指导计算机软件系统开发和维护的工程学科,是一门新兴的边缘学



科。为了使我国的软件产业迅速发展,不断提高软件开发的质量和水平,每一个立志于软件方面的人员,都应该学习和掌握软件工程的基本理论和技术,了解软件经济学,提高软件产业的管理水平,使我国的软件产业能够在国际竞争中尽快地占领一席之地。

在世界上第一台计算机出现之后,就有了程序的概念,可以说程序是软件的前身。而软件是程序、数据及相关文档资料的集合。软件一般由三部分组成:一是程序,它是按事先设计的功能和性能要求执行的指令序列;二是数据,它是使程序能够正常操纵信息的数据结构;三是相关的文档资料,该部分既是软件开发工程中的质量保证,又是软件使用与维护的依据。

那么编写程序等同于开发软件吗?答案是否定的。原因是人们对程序的需求很明确,编写程序所应完成的工作主要是设计算法(即完成指定功能的步骤),然后用某种程序设计语言表达该算法。而开发软件则要复杂得多,事实上编写程序仅仅是开发软件所应完成的工作的一小部分。为了开发出一个符合用户需要、质量合格的软件,软件工程师首先要搞清楚用户面临的问题是什么;接下来要进行可行性研究,分析用户面临的问题是否有行得通的解决方案,并仅在该软件的开发是可行的前提下,才进行实质性的开发工作;然后应该进行需求分析工作,通过与用户的反复交流,搞清楚用户对该软件的具体需求,这些需求是进行软件设计的依据;在编写程序之前需要先进行设计,通常大型软件的设计工作又分成两个阶段进行,先进行总体设计(又称为概要设计),再进行详细设计;编写程序实质上是把设计结果翻译成用某种程序设计语言书写的程序;程序编写出来之后,还需要经过严格的测试过程,软件确实符合用户需求而且质量合格,才能交付给用户使用。

计算机软件的发展已经经历了几十年,人们对软件的认识更加深刻,到目前为止,软件发展经历了三个阶段:

- 程序设计阶段(1946—1956年):采用“个体生产方式”,即软件开发完全依赖于程序员个人的能力水平。

- 程序系统阶段(1956—1968年):由于软件应用范围及规模的不断扩大,个体生产已经不能够满足软件生产的需要,一个软件需要由几个人协同完成,采用“生产作坊方式”。该阶段的后期,随着软件需求量、规模及复杂度的增大,生产作坊的方式已经不能够适应软件生产的需要,出现了所谓的“软件危机”。

- 软件工程阶段(1968年至今):此阶段的主要任务是为了克服软件危机,适应软件发展的需要,而在软件生产中采用“工程化的生产”方式。

从上面的内容可以了解到,软件危机出现在“程序系统”阶段后期,它是指在计算机软件开发和维护过程中所遇到的一系列问题。总的来说,软件危机包括两个方面:一是如何满足用户日益增长的软件需求;二是如何维护数量不断膨胀的已有软件。其具体表现为:

- 不能正确地估计软件开发成本及进度,致使实际开发成本常常高出预算很多。
- 软件产品不可靠,不符合用户的实际需求,甚至无法使用。
- 交付使用的软件不易演化,以至于人们不得不重复开发类似的软件。

- 软件生产率低下,远远满足不了社会发展的需求。
- 软件通常没有适当的文档资料。

以上列举的仅仅是软件危机的一些明显表现,与软件开发和维护有关的问题远不止这些。

产生软件危机的原因很多,除了与软件本身固有的特征有关以外,还与软件开发范型、软件设计方法、软件开发支持以及软件开发管理有关。为了消除软件危机,我们既要有技术措施(方法和工具),又要有必要的组织管理措施。软件工程正是从技术和管理两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

### 1.1.1 软件工程的定义

软件工程(Software Engineering)是在克服 20 世纪 60 年代末所出现的“软件危机”的过程中逐渐形成与发展的。为了克服软件危机,人们从其他产业的工业化生产中得到启示,于是在 1968 年北大西洋公约组织的软件可靠性会议(NATO)上,首次提出了“软件工程”的概念,就是在软件生产中采用工程化的原理和方法,对软件进行规划定义、开发和维护。其宗旨是以较少的投入,在最短的时间内生产出高质量的软件,这种工程化的思想贯穿到软件开发和维护的全过程。在不到 40 年的时间里,人们对软件工程概念的理解由模糊到逐渐清晰和深化,软件工程也从理论和实践两方面都取得了长足的进步。

软件工程是一门指导计算机软件开发与维护的工程学科。它采用工程的概念、原理、技术和方法来开发与维护软件,把正确的管理技术和目前最有效的技术与方法结合起来去开发、生产和维护软件,以达到提高质量、降低成本的目的。

软件工程的方法、工具和过程构成了软件工程的三要素。

软件工程方法是完成软件开发各项任务的技术方法,即为软件开发提供“如何做”的技术。软件工具是为软件工程方法的运用提供自动的或半自动的软件支撑环境。比如计算机辅助软件工程(CASE)系统。软件工程过程是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发这一目的。它规定了完成各项工作任务的工作步骤,即回答软件开发“何时做”的问题。

软件工程就是包含上述方法、工具及过程在内的一些步骤。

由于软件工程是一门新兴的边缘学科,涉及计算机科学、工程科学、管理科学、数学等多学科,研究的范围广,主要研究如何应用软件开发的科学理论和工程技术来指导大型软件系统的开发。因此,现代大型操作系统的开发,如果不采用软件工程的方法是不可想象的。

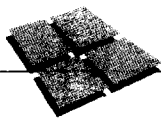
### 1.1.2 软件工程的范畴

为了实现用较低的投入,获得高质量的软件产品这一目标,以满足软件市场日益增长的需要,软件工程需要在方法学、工具与环境、软件管理以及规范与标准等方面进行研究。这也是软件工程的范畴。

#### 1. 软件工程方法学

我们通常把在软件生存周期全过程中使用的一整套技术的集合称为方法学,也称为





范型。在软件工程范畴中,这两个的含义基本相同。由于生存周期中各个阶段的目标和任务不同,因而所用的方法也多种多样。但就方法自身的特征来说可以分为下面几种。

(1)定义说明方法。这种方法的目的是使软件中的操作、操作对象以及其他名词术语都有一个唯一确切的含义,这种含义要为开发者和使用者所理解。定义说明方法一般有两种:一种是形式化方法,这种方法是建立在严密的数学基础上,满足逻辑上的完备性和一致性。比如程序设计语言的文法,常使用有限状态自动机的数学形式来定义。另一种是非形式化定义方法,该方法常常用自然语言来定义,但自然语言容易出现二义性,为尽可能避免二义性,必须遵循形式逻辑学的一些规则,并且要与用户和共同开发者取得一致的理解。

(2)结构化的构造方法。结构化的概念是来源于其他的工程学科,它的基本思想是任何复杂系统都可以分解成一组有限的通用部件,或者说任何一个复杂系统总可以由所选择的通用部件来组成,就像搭积木那样。软件工程中结构化概念的引入是由于结构化程序设计在实践中证明它是保证程序质量的有效途径。因此,人们就把结构化的概念引入到软件生存周期的各个阶段。例如,结构化分析、结构化设计和结构化层次调试等。软件工程中的结构化原理可以归结如下:“用抽象的方法,把事物的固有属性,内部或外部联系抽象成一种具有层次关系的模型,而后按照规定的约束条件进行分解,使得每个层次只能按照规定的基本结构成分来组成。”不同的结构化方法,只是这个原理的具体化而已。结构化方法在实践中是很有效的。

(3)文档资料的描述方法。文档资料是计算机软件产品中不可缺少的一部分。没有这部分,系统就不能进行有效的开发、运行和维护。可是人们往往忽略它。因为这是一件艰苦而枯燥的劳动,开发者往往急于使系统能迅速运行,把精力投入到结构设计和程序设计上,殊不知文档资料不完备或不合乎要求将会造成严重后果。第一,会给系统的可靠性带来严重影响。第二,不能把自己的意图无差错地传达给对方。第三,用户使用时容易出错,即使这个程序编写得非常完善,也不能满足用户的要求。第四,文档资料不合乎要求,将增加开发工作量,延长开发周期,也给读者查询带来困难。因此文档资料要以能完整、确切地描述出所需内容而又不至于繁琐这一总的原则来进行编写。具体编写的原则和方法是:

- 编写文档资料时,必须严格遵守相应的规范和标准。

- 文档资料的详尽程度取决于项目人物的规模、性质、复杂性、使用对象、使用环境和管理水平等因素。具体详尽程度一般由项目负责人根据规范与标准及具体情况来确定。

- 在文档资料中凡是能用图形、表格描述的就不要用文字描述,以保证文档资料简明扼要。

- 在文档资料中,凡是首次应用的专门术语都必须加以定义说明。

### 2. 软件工具和软件支撑环境

为了减少软件生产对人的依存程度,可以通过软件工具的开发来支撑软件人员的工作,这对于提高软件生产效率及可靠性是十分有效的。

软件工具是为软件工程方法的运用提供自动的或半自动的软件支撑环境。它可以是用来帮助开发、测试、分析、维护其他计算机程序及其文档资料的一类程序。如编辑程序、差错程序、诊断程序等。大规模计算机程序及其文档资料的生产所使用的软件工具则是指需求分析工具、设计工具、编码工具、确认工具和维护工具等一类比较复杂的软件工具,是一种自动化系统,在软件开发、维护和管理中,软件工具均起着重要的作用。

需求分析工具的主要功能是辅助分析员把用户所提出的含糊的用户说明,经过分析及一致性、完备性检查后快速生成指导软件设计用的“功能需求说明书”及其相应的文档资料。

设计工具的主要功能是根据输入的功能需求说明,自动设计出一系列软件设计文档,如软件结构说明、模块和接口说明等。

编码工具的主要功能是详细设计所产生的文档,自动生成特定语言编制的程序。例如各种应用程序生成器。

确认工具中的“确认”是指确定程序编码执行和需求说明之间的符合程度。这就包含了各种分析、测试、验证、证明以及排错工作。确认工具的主要功能就是使这一类工作自动化。这一类工具的形式繁多,但可概括为静态分析程序和动态测试程序两大类,每类又分为若干子类。目前能够实现综合确认功能的工具为数很少。

维护工具中的维护是软件生存其中的重要阶段,占用的人力最多,软件费用的比例也最高。这样就迫切需要可行的维护工具,以降低维护费用。目前已有的软件维护工具众多,功能也不尽相同。比如用来协助程序设计课题小组控制源代码;改变源代码的控制系统 SCCS;管理源程序的各个版本;修改需求控制系统 MRCS,它是用来对用户的需求进行跟踪、报告、修改需求的活动,及时控制和协调对软件的更新;等等。

软件工具的种类繁多,形式多样,但都只用于软件生存周期中的某一阶段或某一环节。为了对软件生存周期提供支持,特别提出了建立软件开发支撑环境的课题。

软件开发支撑环境是指在基本硬件与软件的基础上,提供一组能支持软件生存周期的工具,即能支持软件开发、维护、管理和质量控制等各个方面,还能适应多种用户的要求。

软件开发支撑环境一般由数据库、一组工具和一组统一的命令或调用方式三部分组成。其建设规模取决于软件开发项目的一些特征。如开发周期、项目经费、系统寿命、开发人员、开发领域、代码行数和可靠性要求等。目前国际上许多软件公司已经建立起各种用途的软件开发支撑环境。

### 3. 软件管理

软件管理是软件工程中的一项非常重要的工作。事实证明,许多软件项目的开发之所以失败,主要是因为管理不善。目前,软件工程中的维护问题大于开发问题,管理问题大于技术问题。从长远来看,虽然软件开发技术与工具的研究是改善软件生产的根本途径,但是即使软件生产完全自动化了,管理问题仍不容忽视。

软件管理主要包括软件生存周期中的计划管理、成本管理、质量管理和组织管理等。其目的是合理地使用人力、物力资源,以最少消耗来开发软件,保证软件质量,最有效地