高等学校教材

软件工程简明教程

陆惠恩　陆培恩　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等学校教材

软件工程简明教程

陆惠恩　陆培恩　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

软件工程已成为计算机科学的一个重要分支。本书着重从实用角度讲述软件工程的基本概念、原理、方法和工具，系统地介绍目前较成熟的、广泛使用的软件工程技术。

本书内容包括：软件工程概论，需求分析，系统设计与实现（概要设计、详细设计、界面设计与程序设计），软件测试、验证与确认，软件维护，面向对象设计方法和 UML 的使用，软件工程管理技术，软件开发工具与集成化环境，软件开发实例等。每章都有小结供读者复习总结，有习题供选用。

本书可作为应用型本科和高职高专计算机专业的教材，也可供从事计算机软件开发及应用的广大科技人员参考。

图书在版编目(CIP)数据

软件工程简明教程/陆惠恩，陆培恩编著．—北京：电子工业出版社，2005.11
（高等学校教材）
ISBN 7-121-00568-9

Ⅰ．软 … Ⅱ．①陆 … ②陆… Ⅲ．软件工程—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 119004 号

责任编辑：赵家鹏
印　　刷：北京冶金大业印刷有限公司
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：787×1 092　1/16　印张：13　字数：328 千字
印　　次：2005 年 11 月第 1 次印刷
印　　数：4 000 册　　定价：18.00 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。

前　言

随着计算机科学技术的快速发展和计算机应用领域的不断扩大，软件工程技术越来越重要，由于软件的复杂程度的不断增加，对软件的规范化、可维护性和可重用性的要求也越来越高，因而，软件工程学的发展也是非常快的。

为适应当前高等学校应用型本科教学的需要，借鉴高等专科学校培养应用型人才的教学经验，按照应用型本科的教学要求和教材特点，编写了这本《软件工程简明教程》教材。

本书着重从实用角度讲述软件工程的基本概念、原理和方法，系统地介绍目前较成熟、广泛使用的软件工程技术。采用《计算机软件工程规范国家标准汇编 2003》的软件工程术语和规范。

本书编写时除保证应用性强，特别注意语言精炼、易于理解、可读性好的特点。为利于读者掌握重点和巩固学习内容，每章后都有小结和习题。最后一章结合所学课程，通过开发一个小型软件作为软件工程课程设计的示范课题，让读者可参照进行软件的设计和实现。

本书主编陆惠恩。5.1 节、7.1～7.3 节由陆培恩编写，其余由陆惠恩编写。

本书建议理论教学时数为 50 学时(高职、高专教学 40 学时)，另安排课程设计时间两周或实践环节 30 学时。

该书的编写是一个尝试，疏漏和欠妥之处实属难免，欢迎广大师生和读者提出批评和建议。编者的 E-mail 地址为：luhuien@sit.edu.cn。

编者

2005 年 9 月

目　录

第 1 章　概论 …… (1)
1.1　软件工程简述 …… (1)
1.1.1　软件工程发展史 …… (1)
1.1.2　软件危机 …… (2)
1.1.3　软件、软件工程 …… (3)
1.1.4　软件工程的基本原理 …… (3)
1.1.5　软件工程学 …… (4)
1.2　软件过程 …… (6)
1.2.1　软件生存周期 …… (6)
1.2.2　软件开发模型 …… (8)
1.3　软件开发方法 …… (13)
1.3.1　面向数据流设计方法 …… (13)
1.3.2　面向数据结构设计方法 …… (16)
1.3.3　面向对象设计方法 …… (21)
小结 …… (23)
习题 1 …… (24)
第 2 章　需求分析 …… (26)
2.1　需求分析的任务 …… (26)
2.1.1　确定目标系统的具体要求 …… (26)
2.1.2　建立目标系统的逻辑模型 …… (27)
2.1.3　软件需求规格说明 …… (28)
2.1.4　修正系统开发计划 …… (28)
2.1.5　制定初步的系统测试计划 …… (28)
2.1.6　编写用户手册 …… (29)
2.2　需求分析步骤 …… (29)
2.2.1　进行调查研究 …… (29)
2.2.2　分析和描述系统的逻辑模型 …… (29)
2.2.3　复审 …… (30)
2.3　实体-关系图 …… (30)
2.3.1　数据对象 …… (30)
2.3.2　属性 …… (31)
2.3.3　关系 …… (31)
2.3.4　实体-关系图 …… (31)
2.4　数据流图 …… (32)
2.4.1　数据流图的 4 种基本符号 …… (32)
2.4.2　数据流图的几种附加符号 …… (33)
2.4.3　画数据流图的步骤 …… (33)
2.4.4　几点注意事项 …… (34)

2.5 状态转换图 …… (36)
2.5.1 画状态转换图的步骤 …… (36)
2.5.2 状态转换图的符号 …… (36)
2.6 数据字典 …… (37)
2.6.1 数据字典的内容 …… (38)
2.6.2 数据字典使用的符号 …… (38)
2.6.3 数据字典与数据流图 …… (40)
2.7 需求分析图形工具 …… (40)
2.7.1 层次图 …… (40)
2.7.2 Warnier 图 …… (41)
2.7.3 IPO 图 …… (41)
小结 …… (41)
习题 2 …… (42)
第 3 章 系统设计与实现 …… (43)
3.1 概要设计步骤 …… (43)
3.1.1 确定设计方案 …… (43)
3.1.2 软件结构设计 …… (44)
3.1.3 数据文件设计 …… (44)
3.1.4 测试方案设计 …… (44)
3.2 模块和模块化 …… (45)
3.2.1 模块 …… (45)
3.2.2 模块化 …… (45)
3.2.3 模块分割评价标准 …… (46)
3.2.4 模块设计规则 …… (50)
3.3 软件结构设计的图形工具 …… (51)
3.3.1 层次图 …… (51)
3.3.2 结构图 …… (51)
3.4 系统人-机界面设计 …… (53)
3.4.1 人-机界面设计问题 …… (53)
3.4.2 人-机界面设计过程 …… (54)
3.4.3 评估界面设计的标准 …… (54)
3.4.4 界面设计指南 …… (54)
3.5 过程设计 …… (56)
3.6 详细设计工具 …… (56)
3.6.1 流程图 …… (56)
3.6.2 盒图 …… (61)
3.6.3 PAD 图 …… (62)
3.6.4 判定表 …… (64)
3.6.5 判定树 …… (64)
3.6.6 过程设计语言 …… (65)
3.7 结构化设计方法 …… (65)
3.7.1 变换型 …… (66)
3.7.2 事务型 …… (66)

3.8 结构化程序设计 …… (67)
3.8.1 程序设计语言的选择 …… (68)
3.8.2 程序设计风格 …… (69)
3.9 程序结构复杂程度的度量 …… (70)
3.9.1 McCabe 方法 …… (70)
3.9.2 Halstead 方法 …… (72)
小结 …… (72)
习题 3 …… (73)
第 4 章 软件测试、验证与确认 …… (76)
4.1 软件测试目标 …… (76)
4.2 测试方法 …… (76)
4.2.1 静态分析与动态测试 …… (76)
4.2.2 黑盒法与白盒法 …… (77)
4.2.3 测试原则 …… (78)
4.3 测试步骤 …… (79)
4.3.1 模块测试 …… (79)
4.3.2 集成测试 …… (79)
4.3.3 程序审查会和人工运行 …… (80)
4.3.4 确认测试 …… (80)
4.3.5 平行运行 …… (81)
4.4 设计测试方案、实用测试策略 …… (82)
4.4.1 等价类划分法 …… (82)
4.4.2 边界值分析法 …… (82)
4.4.3 错误推测法 …… (83)
4.4.4 逻辑覆盖法 …… (83)
4.4.5 因果图法 …… (86)
4.4.6 实用测试策略 …… (89)
4.4.7 软件调试 …… (90)
4.5 软件验证与确认 …… (91)
4.5.1 软件验证 …… (91)
4.5.2 软件确认 …… (91)
小结 …… (92)
习题 4 …… (92)
第 5 章 软件维护 …… (96)
5.1 维护的定义、特点、过程 …… (96)
5.1.1 维护的定义 …… (96)
5.1.2 维护的特点 …… (97)
5.1.3 维护的过程 …… (98)
5.2 可维护性 …… (101)
5.2.1 决定可维护性的因素 …… (101)
5.2.2 可维护性的度量 …… (101)
5.2.3 如何提高程序的可维护性 …… (103)

小结 …… (103)
习题 5 …… (104)
第 6 章 面向对象方法学 …… (106)
6.1 面向对象技术的概念 …… (106)
6.1.1 面向对象方法学概述 …… (106)
6.1.2 面向对象的概念 …… (107)
6.2 面向对象分析 …… (109)
6.2.1 对象模型 …… (109)
6.2.2 对象抽象的原则 …… (114)
6.2.3 划分主题 …… (114)
6.2.4 建立对象模型的基本方法 …… (115)
6.2.5 动态模型 …… (118)
6.2.6 功能模型 …… (120)
6.2.7 UML 图 …… (121)
6.3 面向对象设计 …… (125)
6.3.1 系统设计 …… (125)
6.3.2 对象设计 …… (128)
6.3.3 面向对象设计的准则 …… (129)
6.3.4 面向对象设计的启发规则 …… (130)
6.4 面向对象系统实现 …… (130)
小结 …… (131)
习题 6 …… (132)
第 7 章 软件工程管理技术 …… (133)
7.1 成本估计技术 …… (133)
7.1.1 代码行技术 …… (133)
7.1.2 任务估算技术 …… (134)
7.2 人员组织 …… (134)
7.2.1 Brooks 定律 …… (134)
7.2.2 软件开发组织的管理结构 …… (134)
7.2.3 程序设计小组的组织 …… (135)
7.3 计划管理 …… (135)
7.3.1 Gantt 图 …… (135)
7.3.2 工程网络技术 …… (136)
7.4 软件配置管理 …… (139)
7.4.1 配置标识 …… (139)
7.4.2 变动控制 …… (139)
7.4.3 配置审计 …… (140)
7.4.4 配置状态报告 …… (140)
7.5 软件质量保证 …… (140)
7.5.1 软件质量的特性 …… (140)
7.5.2 软件质量的保证 …… (141)
7.6 软件工程文件规范 …… (142)

7.6.1 总体要求 …… (143)
7.6.2 可行性研究报告的编写提示 …… (143)
7.6.3 项目开发计划的编写提示 …… (147)
7.6.4 软件需求说明书的编写提示 …… (149)
7.6.5 数据要求说明书的编写提示 …… (151)
7.6.6 概要设计说明书的编写提示 …… (152)
7.6.7 详细设计说明书的编写提示 …… (154)
7.6.8 数据库设计说明书的编写提示 …… (155)
7.6.9 用户手册的编写提示 …… (157)
7.6.10 操作手册的编写提示 …… (159)
7.6.11 模块开发卷宗的编写提示 …… (160)
7.6.12 测试计划的编写提示 …… (162)
7.6.13 测试分析报告的编写提示 …… (163)
7.6.14 开发进度月报的编写提示 …… (165)
7.6.15 项目开发总结报告的编写提示 …… (166)
小结 …… (167)
习题 7 …… (167)
第 8 章 软件开发工具与集成化环境 …… (169)
8.1 CASE 技术 …… (169)
8.1.1 CASE 的基本组成部分 …… (169)
8.1.2 CASE 的软件平台 …… (170)
8.1.3 CASE 的硬件平台 …… (170)
8.2 软件开发工具 …… (170)
8.2.1 软件开发工具的功能 …… (171)
8.2.2 软件开发工具的性能 …… (171)
8.2.3 软件开发工具的分类 …… (172)
8.3 集成化环境 …… (174)
8.3.1 软件工程环境的定义 …… (174)
8.3.2 软件工程环境的分类 …… (175)
8.3.3 软件工程环境的构成和特性 …… (175)
8.3.4 集成化环境 …… (176)
8.3.5 集成化的层次 …… (176)
小结 …… (177)
习题 8 …… (177)
第 9 章 实例——招干考试成绩管理系统 …… (178)
9.1 问题定义 …… (178)
9.2 可行性研究 …… (178)
9.2.1 技术可行性 …… (178)
9.2.2 经济可行性 …… (179)
9.3 需求分析 …… (179)
9.3.1 考生情况分析 …… (179)
9.3.2 成绩输入 …… (179)
9.3.3 录用 …… (179)

9.3.4 输出需求 …… (179)
9.3.5 数据流图和数据字典 …… (180)
9.3.6 IPO 图 …… (180)
9.4 概要设计 …… (181)
9.4.1 数据库结构设计 …… (181)
9.4.2 系统结构设计 …… (181)
9.4.3 测试方案设计 …… (182)
9.5 详细设计 …… (183)
9.5.1 系统界面设计 …… (183)
9.5.2 考前处理 …… (185)
9.5.3 输入设计 …… (186)
9.5.4 成绩处理 …… (188)
9.5.5 录用过程设计 …… (188)
9.5.6 输出设计 …… (189)
9.5.7 测试用例设计 …… (192)
9.6 程序设计提示 …… (193)
9.6.1 进入系统时密码设置 …… (193)
9.6.2 考前处理 …… (194)
9.6.3 成绩输入设计 …… (194)
9.6.4 成绩处理 …… (194)
9.6.5 录用过程设计 …… (194)
9.6.6 初始化程序 …… (194)
9.7 软件测试 …… (195)
习题 9 …… (195)
参考文献 …… (196)

第1章 概 论

1.1 软件工程简述

1.1.1 软件工程发展史

软件工程是随着计算机系统的发展而逐步形成的计算机科学领域中的一门新兴学科。软件工程的发展可分为4个时期。

1.1.1.1 20世纪40年代中期到60年代中期

这个时期计算机硬件从电子管电子计算机发展到晶体管电子计算机，价格昂贵，运算速度低，存储量小。软件通常是规模较小的程序，软件的设计开发者和使用者往往是同一个人。软件设计通常只注意如何节省存储单元、提高运算速度，除了程序清单之外，没有其他任何文档资料。

1.1.1.2 20世纪60年代中期到70年代中期

这个时期计算机硬件发展到集成电路计算机，运算速度和内存容量都相应提高了。出现了“软件作坊”，许多用户不再自己开发软件，而是去“软件作坊”购买软件。随着计算机应用的日益普及，软件需求量急剧增长。用户的需要和使用环境发生变化时，软件可修改性又很差，往往需要重新编制程序，其研制时间很长，不能及时满足用户要求，质量得不到保证。所谓“软件危机”由此开始。

如IBM公司的OS/360系统和美国空军后勤系统，在开发过程中都花费了几千人年的工程量，最后都以失败告终。其中OS/360系统由4000个模块组成，共约100万条指令，花费了5000人年的工程量，经费达数千万美元，结果却失败了。

1968年北大西洋公约组织(NATO)的计算机科学家在联邦德国召开国际会议，正式提出了“软件工程”(Software Engineering)的术语。从此一门新兴的工程学科诞生了。当时“软件工程”还处于学术研究阶段，但已对软件开发产生了巨大影响。著名的例子：1971年IBM公司运用软件工程技术成功地开发了“纽约时报情报库系统”和“空间实验室飞行模拟系统”，而且软件生产率比以前提高了一倍。

1.1.1.3 20世纪70年代中期到80年代

这个时期硬件发展到大规模集成电路计算机，计算机硬件的功能和质量都不断提高。计算机应用不断地扩大，软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的趋势，软件产品供不应求，软件危机日益严重，为了维护软件还要耗费大量的成本。当时美国的统计表明，对计算机软件的投资占计算机软件、硬件总投资的70%，到1985年软件成本大约

占总成本的90%。为了对付不断增长的“软件危机”，软件工程学把软件作为一种产品批量生产。软件工程运用工程学的基本原理和方法来组织和管理软件生产，以保证软件产品的质量和提高软件生产率。

1.1.1.4 20世纪80年代以后

计算机系统发展的第四代不再是单台的计算机和计算机系统，而是计算机软件和硬件的综合效果。由复杂操作系统控制的强大桌面机、广域网和局域网，与先进的应用软件相互配合，计算机体系结构从集中的主机环境转变为分布式的客户机/服务器环境。为此，软件开发的第四代技术可以举例如下：面向对象技术已在许多领域迅速取代了传统的软件开发方法；专家系统和人工智能软件从实验室进入了实际应用，解决了大量的实际问题；人工神经网络软件展示了模式识别与拟人信息处理的前景；并行计算、网络计算机、虚拟现实技术、多媒体技术和现代通信技术使人们可以采用和原来完全不同的方法进行工作。

1.1.2 软件危机

软件危机是指在计算机软件开发和维护时所遇到的一系列问题。软件危机主要包含下面两方面的问题：一是如何开发软件以满足对软件日益增长的需求；二是如何维护数量不断增长的已有软件。

1.1.2.1 软件危机主要表现形式

1. 软件开发成本高，研制进度不能预先估计，用户不满意

由于软件应用范畴越来越广泛，很多软件的应用领域往往是软件开发者不熟悉的，加之开发人员与用户之间信息交流不够，导致软件产品问题太多，研制的进度一再拖延，不能如期完成任务。因而，软件开发成本和进度都与原先的估计相差太大，引起用户不满。

2. 软件产品的质量差，可靠性得不到保证

软件质量无确切的评价标准，软件质量保证技术还没有应用到软件开发的全过程，导致软件产品质量问题频频发生。

3. 软件产品难以维护

早期的程序不注意可读性，不强调可维护性，程序中存在的错误很难改正。用户的需求往往会不断变化，为适应新的要求而维护软件相当困难。当时软件开发不注意保留文档资料，也造成开发和维护的很大困难。

4. 软件发展跟不上硬件的发展和用户的要求

硬件成本逐年下降，软件应用日趋广泛，软件产品“供不应求”，与硬件成本相比，软件成本越来越昂贵。

1.1.2.2 产生软件危机的原因

产生软件危机的原因与软件本身的特点有关，也与软件人员开发时存在的问题有关。

软件是计算机系统中的逻辑部件，软件产品往往规模庞大，给软件的开发和维护带来客观的困难。软件一般要使用8～10年，在这漫长的时间里，很可能出现开发时没有考虑周全的问

题，使运行出现问题，需要及时维护。

软件人员忽视软件需求分析的重要性，轻视软件维护，也是造成软件危机的原因。

1.1.2.3 解决软件危机的途径

目前，计算机的应用日益广泛，世界上发达国家的许多企业将全部投资的 10% 以上用于计算机，而其中 70%以上用于管理方面。但到目前为止，计算机的体系结构在硬件上仍然是冯·诺依曼计算机。硬件的基本功能只能做简单的运算与逻辑判断，主要还是适用于数值计算。对于非数值计算问题，是用编制程序来解决的，因而使软件复杂、庞大，只能由专门的人员来编制软件。假设计算机能实现智能化，计算机能自动进行推理和运算，正确解决用户所提出的问题，那么软件危机就会有根本性的缓解。新一代计算机体系结构的研制可能还需要一段时间。

在目前的计算机硬件条件下，我们要解决以下问题：

(1) 要使用好的软件开发技术和方法；

(2) 要有良好的组织、严密的管理，各类人员要相互配合共同完成任务；

(3) 使用好的软件开发工具，提高软件生产率。

就像机械工具可提高人类的工作能力一样，软件工具可使软件开发工作做得既快又好。如果把各个软件生产阶段使用的工具集合成一个整体，支持软件开发全过程，就构成软件工程支撑环境。软件生产需要开发和使用好的软件支撑环境，为了解决软件危机，需要有技术措施(好的方法和工具)，还要有组织管理措施。软件工程正是从技术和管理这两方面来研究如何更好地开发和维护计算机软件的。

1.1.3 软件、软件工程

软件是指计算机程序及其有关的数据和文档，也包括固化了的程序。

软件的发展大体经历了程序、软件、软件产品等 3 个阶段。早期的程序规模小，随着系统程序的增加，人们把程序区分为系统程序和应用程序，并把它们称为软件，在开发过程中很少考虑到它们的维护问题。当软件需求量大大增加后，人们把软件视为产品，强调软件的“可维护性”，确定了软件开发的各个阶段必需完成的各种规格书、说明书、用户手册等(称为“文档”)。B. Boehm 指出：“软件是程序以及开发、使用和维护所需要的所有文档(document)。”特别当软件成为商品时，文档是必不可少的。没有文档，仅有程序是不能称为软件产品的。

软件工程是软件开发、运行、维护和引退的系统方法。这是国标 GB/T 11457－1995 软件工程术语中定义的。

软件工程是指导计算机软件开发和维护的工程学科。软件工程采用工程的概念、原理、技术和方法来开发与维护软件。软件工程的目标是实现软件的优质高产。

1.1.4 软件工程的基本原理

著名软件工程专家 Boehm 综合有关专家和学者的意见并总结了 TRW 公司多年来开发软件的经验，于 1983 年在一篇论文中提出了软件工程的 7 条基本原理。

(1) 用分阶段的生命周期计划严格管理；

(2) 坚持进行阶段评审；

(3) 实行严格的产品控制；

(4) 采用现代程序设计技术；

(5) 结果应能清楚地审查；

(6) 开发小组的人员应该少而精；

(7) 承认不断改进软件工程实践的必要性。

遵循前 6 条基本原理，能够实现软件的工程化生产；按照第 7 条原理，不仅要积极主动地采纳新的软件技术，而且要注意不断总结经验。在本课程的学习中，读者将体会到这 7 条基本原理的含义和作用。

1.1.5 软件工程学

软件工程学的主要内容是软件开发技术和软件工程管理。其中，软件开发技术包含了软件开发方法学、软件工具和软件工程环境；软件工程管理学包含了软件工程经济学和软件管理学。

1.1.5.1 软件开发方法 (Software Development Methods)

早期的程序设计属于个人活动性质，程序员无统一的方法可循，到了 20 世纪 60 年代后期，兴起了结构程序设计，人们采用结构化的方法来编写程序。

经典的结构程序设计只允许使用顺序结构、条件分支结构和循环结构这 3 种基本结构。这样不仅可改善程序的清晰度，而且能提高软件的可靠性和生产率。随后，人们认识到编写程序仅是软件开发过程中的一个环节。典型的软件开发工作中编写程序所需要的工程量只占软件开发全部工作量的 10%～20%。有效的开发应包括需求分析、软件设计、编写程序等几个阶段，于是形成了“结构化分析”、“结构化设计”、Jackson 方法、Warnier 方法等软件开发方法。到 20 世纪 80 年代又广泛应用了面向对象设计方法。各种软件开发方法的适用范围不尽相同，本书介绍一些比较成熟的目前广泛使用的软件开发方法。

1.1.5.2 软件工具 (Software tools)

为了提高软件设计的质量和生产效率，已发展了许多“帮助开发和维护软件的软件，人们称之为软件工具 (Software tools)，也称软件自动工具 (Automated tools)。

例如，我们要在微机上用某种高级语言开发一个应用软件，往往首先要用编辑程序把源程序输入计算机，然后用编译程序进行编译，如果发现错误，就要重新用编辑程序对源程序进行修改。编译通过后，用连接程序把所有的目标程序，同有关的库程序连接起来，构成一个可执行软件。这里，编辑程序、编译程序、连接程序及支持它们的计算机操作系统都属于软件工具。另外，有测试阶段的测试数据产生器、排错程序、跟踪程序、静态分析工具和覆盖监视工具等，设计阶段和分析阶段也有一些工具。众多的软件工具组成了“工具箱 (Tool box)”或“集成工具 (Integrated tool)”，供软件开发人员在软件生存期的各个阶段根据不同的需要选择使用合适的工具。目前，软件工具发展迅速，许多用于软件分析和设计的工具正在建立，其目标是实现软件生存期各个环节的自动化。

1.1.5.3 软件工程环境(Software Engineering Environment,简称 SEE)

软件开发方法和工具是软件开发的两大支柱,它们之间密切相关。软件开发方法提出了明确的工作步骤和标准的文档格式,这是设计软件工具的基础,而软件工具的实现又将促进软件开发方法的推广和发展。

软件工程环境正是方法和工具的结合。在 1985 年第八届国际软件工程会议上,关于"软件开发环境"的定义是"软件开发环境是相关的一组软件工具集合,它支持一定的软件开发方法或按照一定的软件开发模型组织而成"。

软件开发环境的设计目标是提高软件生产率和改善软件质量。本书将在以后章节中介绍一些常用的软件开发方法、软件工具及软件工程环境。

1.1.5.4 软件工程管理

一个企业如果只有先进的设备和技术,没有完善的管理,是不可能获得应有的经济效益的。软件生产也一样,如果管理不善,是不可能高质量、按时完成任务的。软件工程管理就是对软件工程生存期内的各阶段的活动进行管理。软件工程管理的目的是为了能按预定的时间和费用,成功地完成软件的开发和维护任务,圆满地完成预定的软件开发项目。

软件工程管理包括软件费用管理、人员组织、工程计划管理、软件配置管理等各方面内容。

1. 费用管理

一般来讲,开发一个软件是一种投资,人们总是期望将来获得较大的经济效益。要从经济角度分析,开发一个系统是否划算,从而让使用部门负责人正确地做出是否开发这项系统的决定。我们从软件开发成本、运行费用、经济效益等方面来估算整个系统的投资和回收的数量。

软件开发成本主要表现为人力消耗及相应的开发人员的工资报酬,软件运行费用取决于系统的操作费用和维护费用。其中操作费用包括操作人员的人数、工作时间、消耗的各类物资等项开支,系统的经济效益是指因使用系统而可以节省的费用和增加的收入。

由于运行费用和经济效益两者在软件的整个生存周期内都存在,总的效益和软件生存周期的长度有关,所以,应合理地估算软件的寿命。一般在进行成本/效益分析时一律假设生存周期为 5 年。

2. 人员组织

软件开发不是个体劳动,需要各类人员协同配合,共同完成工程任务,因而应该有良好的组织、周密的管理。

3. 工程计划管理

软件计划是在软件生存周期的早期确定的。在计划实施过程中,需要时,对工程进度应做适当的调整。在软件开发结束后应写出软件开发总结,以便在今后的软件开发工程中能作出更切实际的计划。

4. 软件配置管理

软件工程各阶段所产生的全部文档和软件本身构成软件配置。每当完成一个软件工程步骤,就涉及到软件工程配置,必须使软件配置始终保持其精确性。软件配置管理就是在系统整个生存周期内控制配置的状态和变动,验证配置项的完整性和正确性。

1.2 软件过程

软件过程是为了获得高质量软件所需要完成的一系列任务的框架，它规定了完成各项任务的工作步骤。

ISO 9000 把软件过程定义为："把输入转化为输出的一组彼此相关的资源和活动"。

过程定义了运用方法的顺序，应该交付的文档，开发软件的管理措施，各阶段任务完成的标志。软件过程必须科学、合理，才能获得高质量的软件产品。

本节介绍软件开发、维护全过程应完成的基本任务。

1.2.1 软件生存周期

软件产品从定义开始，经过开发、使用和维护，直到最后被淘汰的整个过程称为软件生存周期。

这如同一个人从出生开始，经过儿童、青年、中年、老年到死亡。在人的一生中，国家和社会对人的负担主要在儿童、青少年时期的培养及老年丧失劳力后的供养。而人从参加工作后就对国家与社会做贡献。贡献越大，人的价值也就越大。同样，软件生存周期中软件的开发要进行投资，消耗价值，当软件交付使用后开始产生价值，软件维护又要消耗价值。软件生存周期中，消耗价值越少，即软件开发与维护所花的费用越低，软件的使用寿命越长，产生的价值就越大，这就是掌握软件工程学的目的。

生存周期是软件工程的一个重要概念。一个软件产品的生存周期可划分为若干个互相区别而又有联系的阶段。把整个生存周期划分为若干个阶段，是实现软件生产工程化的重要步骤。赋予每个阶段相对独立的任务，逐步完成每个阶段的任务，能够简化每个阶段的工作，容易确立系统开发计划，还可明确系统各类开发人员的分工与职责范围，以便分工协作，保证质量。

每一阶段中的工作均以前一阶段的结果为依据，并作为下一阶段的前提。每个阶段都要有技术审查和管理复审，从技术和管理两方面对这个阶段的开发成果进行检查，及时决定系统是继续进行，还是停工或返工。应防止到开发结束才发现先行工作存在的问题，造成不可挽回的损失和失败的浪费现象。每个阶段都进行的复审，主要检查是否有高质量的文档资料，前一个阶段结束了，后一个阶段才能开始。开发单位的技术人员可根据所开发软件的性质、用途及规模等因素决定在软件生存周期中增加或减少相应的阶段。

一个软件产品的生存周期可划分为若干个阶段，这是实现软件生产工程化的重要步骤。

划分软件生存周期的方法有多种，可按软件规模、种类、开发方式、开发环境等来划分生存周期。不管用哪种方法划分，划分的原则是相同的。

软件生存周期划分的原则：

(1) 各阶段工作任务彼此间尽可能相对独立，便于逐步完成每个阶段的任务，能够简化每个阶段的工作，容易确立系统开发计划；

(2) 同一阶段的工作任务性质尽可能相同，这样，有利于软件工程的开发和组织管理，明确系统各类开发人员的分工与职责范围，以便协同工作，保证质量。

软件生存周期一般由软件计划、软件开发和软件运行维护 3 个时期组成。软件计划时期

分为问题定义、可行性研究两个阶段。软件开发时期可分为需求分析、软件设计、测试等阶段。软件交付使用后在运行过程中需要不断地维护，使软件能持久地满足用户的需要。

下面简单介绍软件生存周期各阶段的主要任务。

1. 问题定义

该阶段是软件生存期中最短的阶段。这个阶段要确定系统的目标、规模和基本任务，要有书面报告。这就需要对系统用户和使用单位的负责人进行调查，问题定义报告要征得用户的同意。

2. 可行性研究

该阶段对问题定义阶段确定的系统目标进行全面的分析，探索问题是否有可能解决，具体地确定工程的规模、目标，并估计系统的成本和效益，得出系统是否需要开发的结论，及时中止不值得投资的项目，避免出现不必要的浪费。

可行性研究的任务是对今后的行动提出建议，其目的是确定问题是否值得解决，而不是立即去解决问题。要达到这个目的必须进行客观分析，而且应在尽量短的时间内确定问题是否可行。因而可行性研究是对系统进行简化了的系统分析和设计。首先需进一步分析问题定义，如果问题定义阶段确定的目标、规模正确，应进一步导出系统逻辑模型，经分析后提出几种可能的解法，对每种解法仔细研究其可行性。如果问题定义阶段确定的规模过大，目标难于达到或开发成本过大，收益甚微不值得投资，就应及时建议停止项目开发，避免人力、物力、时间的浪费。

一般来说，每种解决方法的可行性可从以下两方面进行研究。

(1) 经济可行性　根据系统规模目标，确定系统的硬件资源和软件资源的需求规格，估算软件开发成本，分析系统的经济效益是否能超过它的总成本，从经济角度分析系统是否值得投资。

(2) 技术可行性　分析系统是否有某种约束条件，如果有，要全部列出来。根据现有的技术，分析是否能实现系统目标，研究预定的操作方式用户是否能接收。

例如，某市招干考试成绩管理系统，考生分 3 个专业，不同专业考试科目不同。法律专业考政治、语文、法律；行政专业考政治、语文、行政学；财经专业考政治、语文、财经学。每位考生在报名时登记姓名、地址、年龄、报考专业。考生报名后，招干办公室(简称招干办)根据考生报考的专业及考生地址在市区或郊区编排准考证号码和安排考场。考生参加考试后，输入每个考生的各门课的成绩，并统计出每个考生 3 门课考试成绩的总分。按准考证号的顺序打印出考生考试成绩单，并分发给考生。各专业分别将考生按成绩总分从高到低的次序排序，供用人单位决定录取名单。

对本题进行可行性分析：

(1) 技术可行性　有几千名考生报名参加招干考试，若手工计算每位考生 3 门课成绩总分、填写考生成绩单需一式几份(一份给考生，另一份招干办公室留存)；再将考生按成绩总分排序后供录用单位参考。这些工作都是很繁重的，而且手工进行抄写成绩时，要抄写好几份，很容易出错。若将数据输入计算机，虽然需要花费些时间，但根据这些数据由计算机计算总分和按总分排序的速度很快，数据一次输入后可多次使用(输出考生成绩单给考生，考生成绩表供招干办留存；成绩排序后供录用参考；录用后输出录用通知书给考生，录用名单给招干办公室及录用单位)。可见，用计算机建立数据库、开发数据库管理应用系统进行招干考试成绩管理，在技术上是完全可行的。只要系统界面设计合理，系统用户的操作就会易学易用。