高等学校计算机专业教材

GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI

软件工程基础

◉ 陆惠恩 编著

GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI

人民邮电出版社
POSTS & TELECOM PRESS

高等学校计算机专业教材

软件工程基础

陆惠恩　编著

人民邮电出版社

图书在版编目（CIP）数据

软件工程基础/陆惠恩编著. —北京：人民邮电出版社，2005.9
高等学校计算机专业教材
ISBN 7-115-13929-6

Ⅰ. 软… Ⅱ. 陆… 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（2005）第 098857 号

内 容 提 要

本书系统地介绍了软件工程的基本概念、原理、方法及目前较成熟的软件工程技术。内容包括：软件工程概述，软件的可行性研究、需求分析、概要设计、用户界面设计、过程设计、程序设计、软件测试、软件维护等阶段的方法、步骤和文档规范，面向对象方法和统一建模语言（UML），软件开发环境，软件重用，软件质量保证和软件工程管理等。为了使读者更好地学习和掌握有关知识，每章都有小结，并配有适量的例题和习题。

本书的编写力求做到结合实际、注重应用、便于教学，注意内容的新颖、实用和系统性。

本书可作为普通高等学校计算机科学与技术、计算机软件及计算机应用等专业的本科教材，也可供从事计算机应用软件开发和维护的广大科技人员作参考。

高等学校计算机专业教材

软件工程基础

◆ 编　　著　陆惠恩
　责任编辑　张孟玮
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
　邮编　100061　　电子函件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京隆昌伟业印刷有限公司印刷
　新华书店总店北京发行所经销
◆ 开本：787×1092　1/16
　印张：13.75
　字数：323 千字　　2005 年 9 月第 1 版
　印数：1－3 000 册　　2005 年 9 月北京第 1 次印刷

ISBN 7-115-13929-6/TP・4916

定价：19.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223

编 者 的 话

软件工程是计算机科学与技术专业、计算机软件专业及计算机应用等专业的专业课、主干课。本教材讲述软件工程的基本概念、原理和方法，系统地介绍目前流行的和较成熟的软件工程技术。通过理论教学与实践教学的配合，使学生基本掌握结构化方法和面向对象方法等软件开发技术，初步了解软件复用的概念及基于构件的开发方法，对软件工程管理和软件工程环境等内容有总体了解；学习如何系统地、规范地开发和维护软件，规范地书写软件工程的文档资料，合理地安排软件开发、维护的过程，培养和提高软件开发、维护的能力，提高软件开发的效率和质量。

本书针对培养工程技术人才的要求，在编者多年教学实践和科研的基础上，参阅了大量的国内外有关软件工程的教材和资料编写而成。本书的特点如下：

1．语言流畅、深入浅出、详略适当，具有可读性好、应用性强、易于理解的特点。

2．关于软件工程的基本概念、技术、方法等尽可能采用《计算机软件工程规范国家标准汇编 2003》介绍的软件工程术语和规范。

3．引入软件工程的最新技术。例如，较详细地介绍了统一建模语言（UML）在面向对象技术中的使用；统一过程（RUP）、软件复用技术等。

4．每章都列出了主要内容、重点和小结，并配有适量的、经过精选的例题和习题，以利于读者对内容的学习和理解。

5．书中介绍了软件工程各阶段文档书写的规范，使读者在编写文档时，有参考依据。

本课程宜安排在程序设计语言、数据库原理、数据结构等专业课之后，毕业实习、毕业设计之前开设。

本课程的教学重点如下：

（1）软件工程的结构化方法；

（2）面向对象方法；

（3）软件需求分析；

（4）系统结构设计；

（5）软件测试；

（6）软件质量保证。

学习《软件工程基础》，建议理论学习为 45～50 学时，并适当安排实践环节。通过软件开发的实际训练来培养和提高学生开发、维护软件的能力。

实践环节可要求学生完成一个难度适当的软件设计课题。可集中 2～4 周进行课程设计，也可在每章结束时安排实践环节，分阶段逐步完成课题。

本书各章的插图和第 2～8 章中的文档书写规范部分由胡瑞明完成，其余由陆惠恩编写，同济大学计算机科学与技术系博士生导师曹立明教授担任主审，仔细审阅了本书全文，并提出了宝贵的修改意见，作者在此向他表示衷心的感谢！

本书难免存在错误和不足之处，敬请读者批评指正，编者将不胜感激。编者的电子函件地址为：luhuien@sit.edu.cn。

编者

2005 年 6 月

目　录

第 1 章　概述 …… 1
1.1　软件工程简介 …… 1
1.1.1　软件生产的发展 …… 1
1.1.2　软件危机 …… 2
1.1.3　软件工程 …… 3
1.2　软件工程学 …… 4
1.2.1　软件工程学的主要内容 …… 4
1.2.2　软件工程过程 …… 6
1.2.3　软件工程的基本原理 …… 6
1.3　软件生存周期 …… 7
1.4　软件开发模型 …… 8
1.4.1　瀑布模型 …… 9
1.4.2　快速原型模型 …… 10
1.4.3　增量模型 …… 11
1.4.4　喷泉模型 …… 12
1.4.5　螺旋模型 …… 13
1.4.6　统一过程 …… 14
本章小结 …… 15
习题一 …… 16
第 2 章　软件定义与软件计划 …… 18
2.1　软件定义与可行性研究 …… 18
2.1.1　软件定义 …… 18
2.1.2　可行性研究 …… 19
2.1.3　可行性研究的结论 …… 20
2.2　软件工程开发计划 …… 20
2.2.1　软件工程项目概述和实施计划 …… 21
2.2.2　软件开发计划与复审 …… 21
本章小结 …… 22
习题二 …… 23
第 3 章　需求分析 …… 24
3.1　需求分析的任务 …… 24
3.1.1　确定目标系统的具体要求 …… 25
3.1.2　建立目标系统的逻辑模型 …… 26
3.2　结构化分析步骤 …… 27

3.2.1 进行调查研究 …… 27
3.2.2 分析和描述系统的逻辑模型 …… 28
3.2.3 需求分析的复审 …… 28
3.3 需求分析的图形工具 …… 29
3.3.1 实体–关系图 …… 29
3.3.2 数据流图 …… 31
3.3.3 状态转换图 …… 33
3.3.4 IPO 图 …… 34
3.4 数据字典 …… 35
3.4.1 数据字典的内容 …… 35
3.4.2 数据字典使用的符号 …… 36
3.4.3 数据字典与图形工具 …… 37
3.5 需求分析文档 …… 38
3.5.1 软件需求规格说明 …… 38
3.5.2 编写需求分析文档的步骤 …… 38
3.5.3 用户手册编写提示 …… 39
本章小结 …… 39
习题三 …… 40
第 4 章 概要设计 …… 42
4.1 概要设计步骤 …… 42
4.1.1 软件结构设计 …… 42
4.1.2 数据结构及数据库设计 …… 43
4.1.3 系统接口设计 …… 44
4.1.4 设计测试方案 …… 44
4.2 软件结构设计的基本原理 …… 44
4.2.1 模块与信息隐蔽 …… 44
4.2.2 模块化 …… 45
4.2.3 模块的耦合和内聚 …… 47
4.2.4 软件结构设计优化准则 …… 49
4.3 软件结构设计的图形工具 …… 51
4.3.1 层次图 …… 51
4.3.2 结构图 …… 51
4.4 概要设计方法 …… 52
4.4.1 结构化方法 …… 53
4.4.2 面向数据结构的设计方法 …… 55
4.5 代码设计 …… 58
4.5.1 代码设计原则 …… 59
4.5.2 代码种类 …… 59
4.5.3 代码设计方法 …… 61

4.6　数据输入/输出设计 …… 61
4.6.1　输入设计 …… 61
4.6.2　输出设计 …… 62
4.7　数据的安全设计 …… 62
4.8　概要设计文档与复审 …… 63
4.8.1　概要设计说明书 …… 63
4.8.2　概要设计复审 …… 64
4.8.3　数据库设计说明书 …… 65
本章小结 …… 66
习题四 …… 66
第 5 章　详细设计 …… 68
5.1　用户界面设计 …… 68
5.1.1　用户界面设计问题 …… 68
5.1.2　用户界面设计过程 …… 70
5.1.3　用户界面设计的基本原则 …… 70
5.1.4　用户界面设计指南 …… 70
5.2　过程设计 …… 72
5.2.1　流程图 …… 72
5.2.2　盒图 …… 76
5.2.3　PAD 图 …… 77
5.2.4　判定表 …… 79
5.2.5　判定树 …… 79
5.2.6　过程设计语言 …… 80
5.3　详细设计文档与复审 …… 82
5.3.1　详细设计说明书 …… 82
5.3.2　操作手册编写提示 …… 82
5.3.3　详细设计的复审 …… 83
本章小结 …… 84
习题五 …… 84
第 6 章　软件实现 …… 85
6.1　结构化程序设计 …… 85
6.2　选择程序设计语言 …… 86
6.3　程序设计风格 …… 88
6.4　程序设计质量的评价 …… 89
6.5　程序设计文档 …… 90
6.6　软件测试的目标和原则 …… 90
6.6.1　软件测试的目标 …… 90
6.6.2　测试原则 …… 91
6.7　软件测试的方法 …… 91

6.7.1 静态分析与动态测试 …… 91
6.7.2 黑盒法与白盒法 …… 92
6.8 软件测试的步骤 …… 93
6.8.1 模块测试 …… 93
6.8.2 集成测试 …… 93
6.8.3 程序审查会和人工运行 …… 94
6.8.4 确认测试 …… 95
6.8.5 平行运行 …… 96
6.9 设计测试方案 …… 96
6.9.1 等价类划分法 …… 96
6.9.2 边界值分析法 …… 97
6.9.3 错误推测法 …… 97
6.9.4 逻辑覆盖法 …… 98
6.9.5 因果图法 …… 101
6.9.6 实用测试策略 …… 103
6.10 软件调试、验证与确认 …… 104
6.10.1 软件调试 …… 104
6.10.2 程序正确性验证 …… 105
6.10.3 软件确认 …… 106
6.11 软件测试计划和分析报告 …… 106
本章小结 …… 107
习题六 …… 108
第 7 章 软件维护 …… 112
7.1 软件维护过程 …… 112
7.1.1 维护的种类 …… 112
7.1.2 结构性维护与非结构性维护 …… 113
7.1.3 维护的费用 …… 114
7.1.4 维护的困难 …… 114
7.1.5 维护的实施 …… 114
7.1.6 维护的副作用 …… 116
7.2 软件的可维护性 …… 117
7.2.1 决定软件可维护性的因素 …… 117
7.2.2 可维护性的度量 …… 117
7.2.3 提高软件的可维护性 …… 118
本章小结 …… 119
习题七 …… 120
第 8 章 面向对象方法学与 UML …… 121
8.1 面向对象方法概述 …… 121
8.1.1 面向对象方法学的主要优点 …… 122

8.1.2　面向对象的概念 …… 123
8.2　UML 概述 …… 125
8.2.1　UML 的发展 …… 125
8.2.2　UML 的设计目标和内容 …… 126
8.2.3　UML 的语义 …… 127
8.3　UML 图 …… 128
8.3.1　用例图 …… 128
8.3.2　类图和包 …… 129
8.3.3　对象图 …… 133
8.3.4　状态图 …… 133
8.3.5　顺序图 …… 135
8.3.6　活动图 …… 135
8.3.7　协作图 …… 136
8.3.8　构件图 …… 137
8.3.9　部署图 …… 137
本章小结 …… 138
习题八 …… 139
第 9 章　面向对象技术及 UML 应用 …… 140
9.1　面向对象分析 …… 140
9.1.1　面向对象的分析过程 …… 140
9.1.2　面向对象的分析原则 …… 140
9.2　建立对象模型 …… 141
9.2.1　确定对象和类 …… 141
9.2.2　确定类的相互关系 …… 142
9.2.3　划分主题 …… 145
9.3　建立动态模型 …… 147
9.3.1　编写脚本 …… 148
9.3.2　设计用户界面 …… 149
9.3.3　画 UML 顺序图或活动图 …… 149
9.3.4　画状态转换图 …… 149
9.4　建立功能模型 …… 150
9.5　面向对象设计 …… 151
9.5.1　系统设计 …… 152
9.5.2　对象设计 …… 155
9.5.3　面向对象设计的准则 …… 156
9.5.4　面向对象设计的启发规则 …… 156
9.6　UML 的应用 …… 157
9.6.1　UML 模型 …… 157
9.6.2　UML 视图 …… 158

9.6.3 UML 使用准则 …… 160
9.6.4 UML 的扩展机制 …… 161
9.6.5 UML 的应用领域 …… 162
9.7 面向对象系统的实现 …… 162
9.7.1 选择程序设计语言 …… 162
9.7.2 面向对象程序设计 …… 163
9.7.3 面向对象的测试 …… 164
9.8 统一过程 …… 164
9.8.1 RUP 的开发模式 …… 165
9.8.2 RUP 的特点 …… 166
9.8.3 RUP 的要素 …… 166
本章小结 …… 167
习题九 …… 168
第 10 章 软件开发环境 …… 169
10.1 软件开发工具 …… 169
10.2 软件工程环境 …… 172
10.3 CASE 技术 …… 174
本章小结 …… 175
习题十 …… 176
第 11 章 软件重用 …… 177
11.1 可重用的软件成分 …… 177
11.2 软件重用的过程 …… 178
11.2.1 软件重用过程模型 …… 178
11.2.2 开发可重用的软件构件 …… 179
11.2.3 分类和检索软件构件 …… 180
11.2.4 软件重用环境 …… 181
本章小结 …… 181
习题十一 …… 182
第 12 章 软件工程管理 …… 183
12.1 软件工程管理概述 …… 183
12.2 软件规模估算 …… 184
12.2.1 软件开发成本估算方法 …… 184
12.2.2 代码行技术和任务估算技术 …… 185
12.2.3 功能点技术 …… 186
12.2.4 COCOMO 模型 …… 188
12.2.5 程序环行复杂程度的度量 …… 190
12.3 进度计划 …… 191
12.3.1 Gantt 图 …… 191
12.3.2 工程网络技术 …… 191

12.4　人员组织 …… 194
12.5　软件配置管理 …… 196
12.6　软件质量保证 …… 199
12.6.1　软件质量的特性 …… 199
12.6.2　软件质量保证措施 …… 200
12.7　软件工程标准与软件文档 …… 201
12.7.1　软件工程标准 …… 201
12.7.2　软件文档的编写 …… 203
本章小结 …… 204
习题十二 …… 204
附录　选择题参考答案 …… 206
参考文献 …… 207

第1章 概 述

随着计算机应用的日益广泛，计算机软件的开发、维护工作越来越重要。如何以较低的成本开发出高质量的软件？如何开发出用户满意的软件？怎样使所开发的软件在运行中容易维护，以延长软件的使用时间？这些就是软件工程学研究的问题。软件工程是指导计算机软件开发和维护的工程学科。

本章主要介绍软件工程的发展过程，软件危机的形成及如何消除软件危机，软件工程学的基本概念、内容及基本原理。

1.1 软件工程简介

1.1.1 软件生产的发展

自从20世纪40年代电子计算机发明以来，计算机软件随着计算机硬件的发展而逐步发展，软件和硬件一起构成计算机系统。一开始只有程序的概念，后来才出现软件的概念。当软件需求量大大增加后，人们把软件视为产品，确定了软件生产的各个阶段所必须完成的有关计算机程序的功能、设计、编制的文字或图形资料，这些资料称为“文档”。

软件的生产也随着计算机系统的发展而迅速发展，大体经历了程序设计、软件、软件工程及第4代技术等阶段。

1. 程序设计阶段

20世纪40年代中期到60年代中期，电子计算机价格昂贵、运算速度低、存储量小。计算机程序主要是描述计算任务的处理对象和处理规则。早期的程序规模小，程序往往是个人设计、自己使用。设计程序时，通常要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。

2. 软件＝程序＋文档阶段

20世纪60年代中期到70年代中期，采用集成电路制造的计算机运算速度和内存容量大大提高。随着程序的增加，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。随着计算机技术的发展，计算机软件的应用范围也越来越广泛，当软件需求量大大增加后，许多用户去“软件作坊”购买软件。

软件产品交付给用户使用之后，为了纠正错误或适应用户需求的改变，对软件进行的修改，称为软件维护（Software Maintenance）。以前，由于在软件开发过程中很少考虑到它们的维护问题，软件维护的费用以惊人的速度增长，而且不能及时满足用户要求，软件质量得不到保证。所谓“软件危机”由此开始。人们开始重视软件的“可维护性”问题，软件开发采用结构化程序设计技术，规定软件开发时必须书写各种需求规格说明书、说明书及用户手册等文档。

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，讨论软件危机问题，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。

3. 软件工程阶段

20 世纪 70 年代中期到 90 年代，采用大规模集成电路制作的计算机的功能和质量不断提高，个人计算机已经成为大众化商品，计算机应用空前普及。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。为了维护软件要耗费大量的资金。美国当时的统计表明，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进行批量生产，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。软件生产使用数据库、软件开发工具、开发环境等，软件开发技术有了很大的进步，开始采用了工程化开发方法、标准和规范，以及面向对象技术。

4. 第 4 代技术阶段

计算机系统发展的第 4 阶段不再是单台的计算机和计算机程序，而是面向计算机和软件的综合影响。由复杂的操作系统控制的强大的桌面系统，连接局域网和因特网、高带宽数字通信，与先进的应用软件相互配合，产生了综合的效果。计算机体系结构从主机环境转变为分布式的客户/服务器环境。

软件开发的第 4 代技术有了新的发展。与计算机辅助软件工程（CASE）工具和代码生成器结合起来，为许多软件系统提供了可靠的解决方案；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有了实际应用；人工神经网络软件展示了信息处理的美好前景；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开始采用和原来完全不同的方法进行工作。

光计算机、化学计算机、生物计算机和量子计算机等新一代计算机的研制发展，必将给软件工程技术带来一场新的革命。

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。软件危机主要包含以下两方面的问题：

（1）如何开发软件以满足社会对软件日益增长的需求；

（2）如何维护数量不断增长的已有软件。

1. 软件危机的主要表现形式

◆ 软件发展速度跟不上硬件的发展和用户的需求。

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。

◆ 软件成本高，开发进度不能预先估计，用户不满意。

由于软件应用范围越来越广，很多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，导致软件产品不符合要求，不能如期交付。因而，软件开发成本和进度都与原计划相差太大，引起用户不满。

◆ 软件产品质量差，可靠性不能保证。

软件质量保证技术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。

◆ 软件产品可维护性差。

软件设计时不注意程序的可读性，不重视可维护性，程序中存在的错误很难改正。软件需求发生变化时，维护相当困难。

◆ 软件没有合适的文档资料。

软件开发时文档资料不全或文档与软件不一致，使用户不满意，也造成软件维护的很大困难。

2. 软件危机产生的原因

软件危机产生的原因与软件的特点有关，也与软件开发的方式、方法、技术和软件人员本身有关。

◆ 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 5～10 年，在这段时间里，很可能出现开发时没有预料到的问题。例如，系统运行的硬件、软件环境发生了变化、系统需求发生了变化，需要及时地维护软件，使软件可以继续使用。

◆ 软件开发技术落后，生产方式和开发工具落后。

◆ 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。

3. 解决软件危机的途径

目前，计算机的硬件是冯·诺依曼计算机结构。硬件的基本功能只是做简单的运算与逻辑判断，主要适用于数值计算。随着计算机应用的日益广泛，许多企事业单位的计算机，80%以上用于管理方面。对于这样的非数值计算问题，要设计计算机软件来进行处理，因而使软件复杂、庞大。

要解决软件危机问题，需要采取以下措施。

◆ 使用好的软件开发技术和方法。

◆ 使用好的软件开发工具，提高软件生产率。

◆ 有良好的组织、严密的管理，各类人员相互配合共同完成任务。

为了解决软件危机，既要有技术措施（好的方法和工具），也要有组织管理措施。软件工程正是从技术和管理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 软件工程

1. 软件

软件是计算机程序及其有关的数据和文档。

计算机程序是能够完成预定功能和性能的可执行的指令序列；数据是程序能适当处理的信息，具有适当的数据结构；软件文档是开发、使用和维护程序所需要的图文资料。

软件文档（Software Documentation）是以可读的形式出现的技术数据和信息。文档描述或规定软件设计细节，说明软件具备的能力，或为使用软件以便从软件系统得到所期望的结果而提供的操作指令。

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）。”

特别是当软件成为商品时，文档更是必不可少的。没有文档仅有程序，是不能称为软件产品的。

2. 软件工程

软件工程（Software Engineering）是计算机科学中的一个重要分支。GB/T 11457—1995《软件工程术语》对软件工程的定义是：软件工程是软件开发、运行、维护和引退的系统方法。

因而，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。软件工程的目的是在预算范围内，按期交付出用户满意的、质量合格的软件产品。

1.2 软件工程学

1.2.1 软件工程学的主要内容

软件工程学的主要内容是软件开发技术和软件工程管理。

软件开发技术包含软件工程方法学、软件工具和软件工程环境；软件工程管理学包含软件工程经济学和软件管理学。

1. 软件工程方法学

最初，程序设计是个人进行的，只注意如何节省存储单元、提高运算速度。后来，兴起了结构程序设计，人们采用结构化的方法来编写程序。结构程序设计只有顺序结构、条件分支结构和循环结构这三种基本结构。这样不仅改善了程序的清晰度，而且能提高软件的可靠性和生产率。

后来，人们逐步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的工程量只占软件开发全部工作量的 10%～20%。软件开发工作应包括需求分析、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、面向数据结构的 Jackson 方法、Warnier 方法等传统软件开发方法，20 世纪 80 年代广泛应用了面向对象设计方法。

软件开发方法学是编制软件的系统方法，它确定软件开发的各个阶段，规定每一阶段的活动、产品、验收的步骤和完成准则。

软件工程方法学有三个要素：方法、工具和过程。

- 方法：完成软件开发任务的技术方法。
- 工具：为方法的运用提供自动或半自动的软件支撑环境。
- 过程：规定了完成任务的工作阶段、工作内容、产品、验收的步骤和完成准则。

各种软件开发方法的适用范围不尽相同。目前使用得最广泛的软件工程方法学是传统方法学和面向对象方法学。

（1）传统方法学

采用结构化技术，包括结构化分析、结构化设计和结构化实现，来完成软件开发任务。把软件开发工作划分成若干个阶段，顺序完成各阶段的任务；每个阶段的开始和结束都有严

格的标准；每个阶段结束时要进行严格的技术审查和管理复审。传统方法学先确定软件功能，再对功能进行分解，确定怎样开发软件，然后再实现软件功能。

（2）面向对象方法学

面向对象方法学是把对象作为数据和在数据上的操作相结合的软件构件。用对象分解取代了传统方法的功能分解。把所有对象都划分成类，把若干个相关的类组织成具有层次结构的系统。对象之间通过发送消息相互联系。

2. 软件工具

软件工具（Software Tools）是指为了支持计算机软件的开发和维护而研制的程序系统。使用软件工具的目的是提高软件设计的质量和生产效率，降低软件开发、维护的成本。

软件开发工具用于软件开发的整个过程。例如，需求分析工具用类生成需求说明；设计阶段需要使用编辑程序、编译程序、连接程序，有的软件能自动生成程序等；在测试阶段可使用排错程序、跟踪程序、静态分析工具和监视工具等；软件维护阶段有版本管理、文档分析工具等；软件管理方面也有许多软件工具。众多的软件工具组成了“工具箱（Tool Box）”或“集成工具（Integrated Tool）”。软件开发人员在软件生产的各个阶段可根据不同的需要选择合适的工具使用。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产各阶段的自动化。

3. 软件工程环境

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（Software Engineering Environment，SEE）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。本书将在以后章节介绍一些常用的软件开发方法、软件工具及软件工程环境。

计算机辅助软件工程（Computer Aided Software Engineering，CASE）是一组工具和方法的集合，可以辅助软件生存周期各阶段进行软件开发活动。

CASE 是多年来在软件开发管理、软件开发方法、软件开发环境和软件工具等方面研究和发展的产物。CASE 吸收了 CAD（计算机辅助设计）、软件工程、操作系统、数据库、网络和许多其他计算机领域的原理和技术。因而，CASE 领域是一个应用、集成和综合的领域。其中，软件工具不是对任何软件开发方法的取代，而是对方法的辅助，它旨在提高软件开发的效率和软件产品的质量。

4. 软件工程管理

软件工程管理就是对软件开发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，成功地生产软件产品。软件工程管理的任务是有效地组织人员、按照适当的技术、方法，利用好的工具来完成预定的软件项目。

软件工程管理的内容包括软件费用管理、人员组织、工程计划管理、软件配置管理等方面的内容。

（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。从经济角度分析，开发一个软件系统是否划算，是软件使用单位负责人决定是否开发这个项目的主要依据。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报情况。

软件开发成本主要包括开发人员的工资报酬、开发阶段的各项支出。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、消耗的各类物资等开支。系统的经济效益是指因使用新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使用期内都存在，总的效益和软件使用时间的长短有关。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，一律假设软件使用期为 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

软件工程计划是在软件开发早期确定的。在计划实施过程中，在需要时，应对工程进度作适当的调整。在软件开发结束后应写出软件开发总结，以便今后能制定出更切合实际的软件开发计划。

（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的整个开发、运行和维护阶段内控制配置的状态和变动，验证配置项的完全性和正确性。

1.2.2 软件工程过程

国际标准化组织（International Standards Organization，ISO）是世界性的标准化专门机构。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。

软件工程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。

软件开发过程（Software Development Process）是把用户要求转化为软件需求，把软件需求转化为设计，用代码来实现设计、对代码进行测试，完成文档编制并确认软件可以投入运行性使用的过程。

软件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和各阶段任务完成的标志。

软件过程是软件工程方法学的三个要素（方法、工具和过程）之一。软件过程必须科学、合理，才能获得高质量的软件产品。

1.2.3 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）用分阶段的生存周期计划进行严格的管理。