

 高等教育规划教材

# 软件工程基础

宋雨 编著



免费提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社  
CHINA MACHINE PRESS

高等教育规划教材

# 软件工程基础

宋 雨 编著



机械工业出版社

本书理论联系实际,按照软件工程的基本原理,从实用的角度介绍了软件需求分析方法、软件设计方法、软件的编程实现要求、软件的测试及维护方法以及软件项目管理方法,而这些也是软件工程学科的基本内容。书中第7章还给出了若干经典案例供读者学习。为促进学习效果,第8章给出了140个精选软件工程设计题目及其功能要求供读者作为软件工程课题选用,第8章还给出了课程设计的评分标准供教师参考。每章都有结合实际的案例以及供读者练习的相应习题。

本书是软件工程的入门教材,内容通俗、易懂,注重趣味性、故事性和情节性。本书适合作为高等学校计算机相关专业的教学用书,也可作为对软件工程学科感兴趣的工程技术人员的参考用书。

本书配套授课电子课件,需要的教师可登录 [www.cmpedu.com](http://www.cmpedu.com) 免费注册,审核通过后下载,或联系编辑索取(QQ: 2850823885,电话: 010-88379739)。

### 图书在版编目(CIP)数据

软件工程基础 / 宋雨编著. —北京: 机械工业出版社, 2016.3

高等教育规划教材

ISBN 978-7-111-52511-0

I. ①软… II. ①宋… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2016)第 021357 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 张 恒 责任编辑: 张 恒

责任校对: 张艳霞 责任印制: 李 洋

三河市宏达印刷有限公司印刷

2016 年 3 月第 1 版·第 1 次印刷

184mm × 260mm · 20.25 印张 · 499 千字

0001—3000 册

标准书号: ISBN 978-7-111-52511-0

定价: 49.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88361066

机工官网: [www.cmpbook.com](http://www.cmpbook.com)

读者购书热线: (010) 68326294

机工官博: [weibo.com/cmp1952](http://weibo.com/cmp1952)

(010) 88379203

教育服务网: [www.cmpedu.com](http://www.cmpedu.com)

封面无防伪标均为盗版

金书网: [www.golden-book.com](http://www.golden-book.com)



# 出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师能提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

# 前 言

人类已进入大数据时代，这是一个信息爆炸的时代，这都应该归因于计算机处理能力的提高，而计算机中起主导作用的是软件。稍加注意就会发现，软件无处不在，软件工程无处不在。毫不夸张地说，软件在技术进步方面起到了决定性的作用，国家财政收入中与软件产业的相关度越来越高，软件产业收入所占的比例也越来越高，软件及软件产业发展的速度超出人们的想象，2014年我国的软件产业总收入达到了3.7万亿元。与此同时，与软件产业紧密相关的软件工程也得到了很大的重视，软件工程作为一门学科虽然只有40多年的时间，但从它诞生之日起就显示出极强的生命力。

为适应软件工程学科快速发展，2011年国务院学位委员会及教育部将软件工程设立为国家一级学科，这既为软件工程学科发展指明了方向，也说明软件工程的地位和作用十分重要。软件工程涉及的内容广泛、丰富，无论广度还是深度，其他学科都难以覆盖。

本书作者长期从事软件工程教学和科研，在参考了大量同类文献之后，按学生的接受程度、学科的基本内容和要求编写了本书。本书是软件工程的入门教材，适合初学者作为软件工程课程的教材使用，也适用于对软件工程学科感兴趣的读者阅读。软件工程是快速发展的学科，因而，虽然本书是基础教程但在书中也尽量对发展中的新技术有所反映。

本书的特点如下。

1) 力求通俗、易懂，尽量使枯燥的内容变得有趣，因而在叙述软件工程技术和方法上增加了趣味性、故事性和情节性。

2) 尽量将难点分散，避开高深的知识，并使全书在叙述上有节奏感，增强初学者学习的信心。

3) 理论结合实际，每一章都有案例，使理论不抽象、技术能“落地”。

4) 增强实用性。书中既有案例又列出了大量的实际题目，增强读者学习的牢固性，使其在其他学科的学习和实践中仍然有参考意义。

全书共8章，第1章是软件工程概述，介绍了软件的定义、起源和分类以及软件工程的产生和定义，讨论了软件生命周期的概念，叙述了常见的软件开发模型。第2章阐述了软件需求分析方法，包括结构化分析方法、原型化分析方法、面向对象建模及UML方法，以及软件需求规约说明书(SRS)的构造和要求。第3章是软件系统的设计，介绍了软件设计的基本原理，讲述了结构化设计方法和面向对象的设计方法，简要介绍了一些新方法。第4章是软件的编程实现，包括编程语言的选择、分类和编程要求，介绍了常用的面向对象的编程语言。第5章是软件的测试及维护，叙述了软件测试的基本原理、测试用例设计及面向对象的测试方法，讲述了软件维护的基本原理和软件再工程的内容。第6章是软件项目管理，包括软件范围的确定、软件资源的考虑、软件成本估算模型和技术，以及如何使用甘特图和PERT

图来安排软件工程项目进度。第 7 章是从公开资料中整理而成的典型软件项目案例，具有很高的指导价值。第 8 章是软件工程课程设计的内容、基本要求、考核标准以及交付文档的要求和格式，这一章给出了 140 个具有实际意义的软件工程课程设计题目，可作为课程设计的目标、功能和性能要求。

本书配有习题解答、电子教案等教辅材料。使用本书的课程建议授课学时为 40~48 学时，课程设计可安排 2 周。

本书由华北电力大学宋雨编著。在编写过程中得到了多位同仁的支持和帮助，在这里一并表示感谢。由于时间仓促，书中难免存在不妥之处，请读者批评指正，并提出宝贵意见。

编 者

# 目 录

## 前言

第 1 章 软件工程概述	1	2.3.2 面向对象建模	54
1.1 软件分类和演化	1	2.3.3 统一建模语言 UML	63
1.1.1 软件的起源和分类	1	2.4 需求规约说明书 (SRS)	67
1.1.2 软件工程的产生和发展	4	2.4.1 SRS 的内容	67
1.2 软件的生命周期	8	2.4.2 SRS 的作用	68
1.2.1 计划阶段	9	2.4.3 SRS 的特征	69
1.2.2 分析和定义阶段	10	2.4.4 SRS 的构造原则	70
1.2.3 设计阶段	11	2.4.5 SRS 的评审	71
1.2.4 实现阶段	12	2.5 案例: 图书馆系统的软件需求	
1.2.5 测试阶段	13	分析	72
1.2.6 运行和维护阶段	14	2.5.1 确定系统参与者	72
1.3 软件开发模型	15	2.5.2 开发系统场景	73
1.3.1 瀑布模型	15	2.5.3 绘制系统用例图	73
1.3.2 演化模型	16	2.5.4 描述用例	74
1.3.3 螺旋模型	17	2.6 小结	75
1.3.4 喷泉模型	18	2.7 习题	76
1.3.5 其他模型	18	第 3 章 软件系统的设计	80
1.4 实用案例	23	3.1 软件设计的基本原理	80
1.4.1 出卷系统的开发模型选择	23	3.1.1 软件设计的概念和原则	83
1.4.2 住宅安全系统 SafeHome 的开发		3.1.2 软件概要设计	90
模型选择	25	3.1.3 软件详细设计	97
1.5 小结	26	3.2 结构化设计方法	107
1.6 习题	26	3.2.1 软件结构图的组成	108
第 2 章 软件需求分析	29	3.2.2 软件结构图的画法	109
2.1 结构化需求分析方法	29	3.3 面向对象的设计方法	115
2.1.1 数据流图及其画法	30	3.3.1 面向对象设计过程	115
2.1.2 数据词典及其描述	38	3.3.2 面向对象设计方法	118
2.1.3 功能说明	41	3.4 其他设计方法介绍	130
2.2 原型化分析方法	44	3.4.1 面向方面程序设计	130
2.2.1 开发模型	45	3.4.2 面向 Agent 的设计方法	131
2.2.2 快速原型技术介绍	48	3.4.3 泛型程序设计	131
2.2.3 用户界面开发	51	3.4.4 面向构件的技术	132
2.3 面向对象建模及 UML 方法	51	3.4.5 敏捷方法	132
2.3.1 面向对象基本概念	51	3.4.6 Rational 统一过程	132

3.4.7 功能驱动开发模式 FDD .....	132	5.1.1 软件测试目标和原则 .....	168
3.4.8 极端编程 .....	133	5.1.2 软件的可测试性 .....	169
3.5 实用案例 .....	133	5.1.3 软件测试的步骤和策略 .....	169
3.5.1 SafeHome 软件的结构设计 .....	133	5.2 测试用例设计 .....	179
3.5.2 基于 UML 的网络管理平台的 分析与设计 .....	136	5.2.1 白盒测试 .....	179
3.6 小结 .....	140	5.2.2 黑盒测试 .....	183
3.7 习题 .....	141	5.2.3 人工测试 .....	185
<b>第 4 章 软件的编程实现</b> .....	147	5.2.4 调试 .....	186
4.1 编程语言的选择和分类 .....	147	5.3 面向对象的测试 .....	188
4.1.1 程序设计语言的分类 .....	147	5.3.1 测试策略 .....	188
4.1.2 机器语言 .....	148	5.3.2 类测试方法 .....	189
4.1.3 汇编语言 .....	150	5.4 软件维护的基本原理 .....	193
4.1.4 高级语言 .....	150	5.4.1 软件维护的分类 .....	193
4.1.5 非过程语言 .....	151	5.4.2 影响软件可维护性的特性 .....	195
4.2 编程要求 .....	151	5.4.3 提高软件可维护性的方法 .....	198
4.2.1 程序语句结构的构成原则 .....	152	5.5 软件再工程 .....	201
4.2.2 程序可读性和易理解性的要求 .....	153	5.5.1 业务过程再工程模型 .....	201
4.2.3 数据说明的要求 .....	155	5.5.2 软件再工程过程模型 .....	202
4.2.4 输入和输出应遵守的原则 .....	155	5.5.3 软件的逆向工程 .....	203
4.3 面向对象的编程语言介绍 .....	156	5.5.4 软件重构 .....	205
4.3.1 Smalltalk 语言 .....	156	5.5.5 软件的正向工程 .....	205
4.3.2 Eiffel 语言 .....	156	5.5.6 软件再工程的成本-效益分析 .....	206
4.3.3 C++ 语言 .....	156	5.6 案例: 微软公司的软件测试 .....	207
4.3.4 Delphi 语言 .....	157	5.6.1 软件测试人员的组成及任务 .....	207
4.3.5 Java 语言 .....	158	5.6.2 软件测试应考虑的问题 .....	209
4.3.6 C# 语言 .....	158	5.6.3 对软件中 bug 的处理 .....	209
4.4 案例: 网上招聘系统软件 编程规范 .....	158	5.6.4 采用的软件测试方法及测试工具 .....	210
4.4.1 编码格式规范 .....	158	5.6.5 主要的软件测试文档 .....	212
4.4.2 编码命名规范 .....	160	5.7 小结 .....	213
4.4.3 程序中的声明规范 .....	161	5.8 习题 .....	215
4.4.4 程序语句规范 .....	162	<b>第 6 章 软件项目管理</b> .....	218
4.4.5 代码注释规范 .....	162	6.1 软件范围的确定 .....	218
4.4.6 存放代码的目录规范 .....	164	6.1.1 获取确定软件范围所需的信息 .....	218
4.5 小结 .....	166	6.1.2 软件范围的具体内容及描述要求 .....	219
4.6 习题 .....	166	6.2 软件资源的考虑 .....	219
<b>第 5 章 软件的测试及维护</b> .....	168	6.2.1 人力资源 .....	220
5.1 软件测试的基本原理 .....	168	6.2.2 可复用软件资源 .....	220
		6.2.3 环境资源 .....	221
		6.3 软件成本的估算 .....	221



6.3.1	软件估算模型	222	7.4.1	系统设计目标	260
6.3.2	软件估算方法	232	7.4.2	迭代开发过程	260
6.3.3	面向对象软件项目的估算	236	7.4.3	实现细节考虑	261
6.3.4	软件自行开发或购买的决策	237	7.5	富士通的基于 ZooKeeper 和 Storm 的 车载流式计算框架	262
6.4	开发进度的安排	238	7.5.1	项目背景介绍	262
6.4.1	使用甘特图安排软件工程 项目进度	238	7.5.2	软件设计策略	262
6.4.2	使用 PERT 图安排软件工程 项目进度	240	7.5.3	案例成功要点	263
6.4.3	两种图相结合安排软件工程 项目进度	242	7.6	支付宝无线统一测试平台	264
6.5	案例：会计信息系统软件 成本估算	242	7.6.1	研发模式及实践经验	264
6.5.1	根据已知条件选择估算方法或 模型	243	7.6.2	测试平台的模块结构介绍	265
6.5.2	根据选定的方法或模型进行计算	243	7.6.3	核心模块的功能描述	266
6.5.3	使用其他方法或模型进行 交叉检验	244	7.7	习题	268
6.6	小结	244	<b>第 8 章</b>	<b>软件工程课程设计</b>	269
6.7	习题	245	8.1	基本要求及考核标准	269
<b>第 7 章</b>	<b>软件工程项目案例</b>	251	8.1.1	课程设计的内容	269
7.1	美国航空航天局的太空机器人 系统	251	8.1.2	课程设计的考核	303
7.1.1	以用户为中心的迭代设计过程	251	8.2	交付文档要求及格式	307
7.1.2	面对的技术挑战	252	8.2.1	《软件计划》编写格式及要点	308
7.1.3	太空机器人系统的测试	252	8.2.2	《需求规约说明书(SRS)》编写格式及 要点	309
7.2	乐视网 TV 大数据平台	253	8.2.3	《软件设计说明书》编写格式及 要点	309
7.2.1	问题的提出	253	8.2.4	《软件测试计划》编写格式及 要点	310
7.2.2	平台构建过程及采用的工具	253	8.2.5	《软件测试分析报告》编写格式及 要点	310
7.3	有道云笔记云端架构	257	8.2.6	《开发进度报告》编写格式及 要点	311
7.3.1	数据结构的设计	257	8.2.7	《用户手册》编写格式及要点	311
7.3.2	服务框架的设计	258	8.2.8	《操作手册》编写格式及要点	312
7.3.3	架构的重构和升级	258	8.2.9	《软件开发总结报告》编写格式及 要点	312
7.4	网易广告投放系统	259	<b>参考文献</b>		313

# 第1章 软件工程概述

现在是电子信息时代，大街上的年轻人头上戴着耳机听着什么匆匆走过；咖啡厅、地铁、医院、餐厅、学校、排队等候办事的人群中、甚至公交车上，不少人都在低着头用手机上网、看微信或玩游戏；当我们坐在家玩电脑或上网查资料的时候，当我们到派出所办理户口或身份证的时候，当我们到医院检查身体或看病的时候，当我们到银行办理业务的时候，当我们去车站或机场购票的时候，等等这一切的一切，是依靠什么顺畅运行的呢？答案是软件！软件是什么，它是怎样发展起来的，没有它行不行，软件工程又是什么呢？本章内容将会回答你这些问题。

## 1.1 软件分类和演化

软件工程是围绕软件的开发、维护、管理、质量保证等一系列活动而展开的，本节将介绍什么是软件，它是如何分类，又是如何发展的。

### 1.1.1 软件的起源和分类

#### 1. 软件起源

软件一词起源于程序，是20世纪60年代出现的概念，也是一个不断发展的概念，软件的形成和发展与计算学科的发展紧密相关，现在计算学科已经发展为计算机科学与技术以及软件工程两大学科。不同时期对软件的解释也有所不同，这是由于软件的功能、模块、开发方式以及使用方式在不断发生变化，因而对它的解释也在发展变化。

#### 2. 软件定义

软件是相对于硬件而言的，以解释性的定义为主。

美国软件工程专家罗杰·普雷斯曼（Roger S. Pressman）（如图1-1所示）将软件解释为当机器执行时能提供所要求功能和性能的程序，能使程序有效处理信息的数据结构以及描述操作和使用程序的文档，简单地说就是软件由程序、数据和文档组成。

电子与电气工程师协会（IEEE）将软件定义为计算机程序、方法、规则、相关文档及在计算机中运行时所需的数据。

英国软件工程专家兰·萨默维尔（Ian Sommerville）（如图1-2所示）认为软件是一个系统，通常由若干程序、用于建立这些程序的配置文件、描述系统结构的系统文档、解释如何使用系统的用户文档以及供用户下载最新产品信息的Web站点组成。

一些学者认为软件由程序及其文档组成，其中，程序是计算机任务的处理对象和处理规则的描述，文档是为了理解程序所需的阐述性资料。



图 1-1 罗杰·普雷斯曼



图 1-2 兰·萨默维尔

不难理解，软件其实是知识的载体，它包括的内容和范围很广，一般认为执行指令的计算机程序以及与之相关的文档、数据、影视资料、方法、规则、网页及其链接等都可算作软件。

### 3. 软件分类

从软件的起源和定义可以看出，软件具有 3 层含义：一是个体含义，指计算机系统内的程序、文档和数据；二是整体含义，指在特定计算机系统中所有上述个体含义下的软件的总称；三是学科含义，指在研究、开发、维护以及使用前述含义下的软件所涉及的理论、方法、技术所构成的学科。

软件的种类很多，随着复杂程度的增加，软件分类也在发展变化，界限越来越不明显，按软件的作用，可分为以下 4 类。

#### (1) 系统软件

是指由生产厂家配置，服务于其他系统元素的软件。如操作系统、汇编程序、编译程序、数据库管理系统、计算机通信及网络软件等。如果没有这些软件，计算机、网络、设备等很难发挥功能，甚至无法工作。

#### (2) 应用软件

是指在系统软件的基础上，为解决特定领域问题而开发的软件。这类软件很多，如用于电信、金融、电力、公安、交通管理、招生、考试、录取等领域的专用软件，企事业单位生产、工作、管理、服务的各种事务类软件，监视、分析和控制正在发生的现实世界事件的各种实时软件，各类科学和工程软件，用于工业、民用或军事上的各种功能的与设备融为一体的嵌入式软件，个人计算机软件，手机上的各种实用软件，基于 Web 的软件，儿童玩具中的软件，人工智能软件，等等。

#### (3) 工具软件

用于辅助和支持开发及维护的应用软件，以提高软件开发质量和生产率的软件都可称为工具软件。如软件进度安排软件、软件成本估算工具、软件需求分析工具、软件设计工具、软件编码工具、测试工具、系统维护工具、管理工具等，这些软件工具不仅能完成指定的工作，还能进行完备性、一致性、正确性、安全性、追溯性等方面的检查，完成一些人工所不

能完成的事情。

#### (4) 可重用软件

是指通过修改、剪裁等手段应用于新软件的软件，可重用软件并不仅指代码，也可以是软件设计、规范、数据、测试用例，甚至概念等，可以把它们统称为可重用软件构件，目前，常见的各种图形用户界面一般是使用可重用软件构件创建的，这些构件涉及图形窗口、下拉菜单和各种交互机制。建造用户界面所需要的数据结构和处理细节包含在一个由界面构件所组成的可重用构件库中，开发人员通过可重用构件来开发新的软件。

若按功能对软件进行分类，则可划分为系统软件、支撑软件和应用软件三大类；若按规模进行划分，可将软件分为 7 类，如表 1-1 所示；若按软件工作方式划分，则可分为实时处理软件、分时软件、交互式软件和批处理软件 4 类；若按软件服务对象的范围进行划分，则可分为项目软件和产品软件 2 类；若按软件的应用领域进行划分也可将软件分为 7 类，如表 1-2 所示；还可按其他方式对软件进行划分，如按软件的使用频度进行划分、按软件失效的影响进行划分等。

表 1-1 按规模对软件分类

类别	参加人员数	研制期限	产品规模(源程序行数)
微型	1	1~4 周	0.5 千行
小型	1	1~6 月	1~2 千行
中型	2~5	1~2 年	5~50 千行
大型	5~20	2~3 年	50~100 千行
甚大型	100~1000	4~5 年	1 兆行 (=1000 千行)
极大型	2000~5000	5~10 年	1 兆~10 兆行
超大型	具有互联网规模	长期、持续演化	10 兆~10 亿行或更大

表 1-2 按应用领域对软件分类

类别	用途	特点	例子
系统软件	一整套服务于其他程序的软件	和计算机硬件大量交互；多用户大量使用；需要调度、资源共享和复杂进程管理的同步操作；复杂的数据结构以及多种外部接口	编译器、编辑器、文件管理软件、操作系统构件、驱动程序、网络软件、远程通信处理器等
应用软件	解决特定业务需要的独立应用软件	处理商务或技术数据，以协助业务操作、管理或技术决策	传统数据处理应用程序、超市交易处理软件、实时制造过程控制软件等
工程和科学软件	具有明显“数值计算”算法的软件	不仅仅局限于传统的数值算法，有些工程和科学软件甚至具有系统软件的特征	从天文学到火山学、从自动应力分析到航天飞机轨道动力学、从分子生物学到自动制造业，几乎涵盖了所有的应用领域
嵌入式软件	存在于某个产品或系统中，可实现和控制面向最终使用者和系统本身的特性和功能的软件	可以执行有限但难于实现的功能，提供重要的功能和控制能力	导弹、火箭等武器控制系统，汽车中的燃油控制、仪表盘显示、刹车系统等汽车电子功能，各种电子产品、医疗设备，微波炉、洗衣机、电冰箱的按键控制等
产品线软件	为多个不同用户的使用提供特定功能的软件	关注有限的特定专业市场及大众消费品市场	库存产品控制软件、文字处理软件、电子制表软件、电脑绘图软件、多媒体工具软件、娱乐软件、数据库管理软件、个人及公司财务应用软件
Web 应用软件	是一类以网络为中心的软体	涵盖了宽泛的应用领域	可以是一组超文本链接文件，也可以是复杂的计算环境
人工智能软件	应用于专门场合或领域	利用非数值算法解决计算和直接分析无法解决的复杂问题	机器人、专家系统、模式识别、人工神经网络、定理证明和博弈等

软件产品包括两类，通用软件产品和定制软件产品。它们的区别在于，通用软件产品的描述由开发者自己完成，而定制软件产品的描述通常由客户给出，开发者必须按客户要求开发。因而，各类数据库软件、字处理软件、绘图软件、工程管理软件等属于通用软件产品，而特定的业务处理系统、电子设备的控制系统、空中交通管制系统等属于定制软件产品。

## 1.1.2 软件工程的产生和发展

### 1. 软件工程的产生

20 世纪 40 年代中期，美国宾夕法尼亚大学约翰·莫克莱（John Mauchly）和普雷斯帕·埃克特（J.Presper Eckert）（见图 1-3）研制出世界上第一台电子数字计算机 ENIAC。1947 年，数学家冯·诺依曼（J.Von Neumann，如图 1-4 所示）针对 ENIAC 提出了 EDVAC（Electronic Discrete Variable Automatic Computer）方案，在该方案中首次提出了“存储程序”的概念，图 1-5 所示是 1951 年实现的 EDVAC。1949 年，由英国皇家科学院院士莫里斯·威尔克斯制造的世界上第一台实现冯·诺依曼“存储程序”思想的计算机 EDSAC（Electronic Delay Storage Automatic Calculator）问世。从冯·诺依曼提出存储程序概念至今的时间中，软件发生了很大变化，其发展可用表 1-3 概括。



图 1-3 莫克莱和埃克特



图 1-4 冯·诺依曼

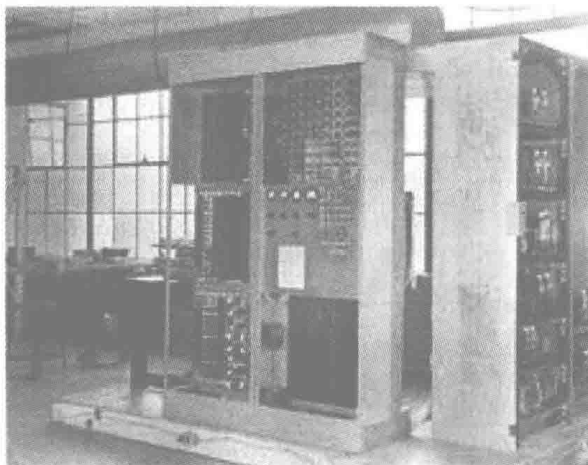


图 1-5 EDVAC (1951)



表 1-3 软件的演化历程及特征

阶段划分	阶段名称	时间	对软件的解释	软件开发方法	决定软件质量的因素	硬件特征	软件特征及技术
第一阶段	程序设计阶段	20 世纪 50 年代	程序	个人	个人程序设计技术	价格昂贵, 存储容量小, 工作可靠性差	软件不受重视
第二阶段	程序系统阶段	20 世纪 60 年代	程序、说明文档	“软件作坊”式的小组	小组技术水平	降价, 速度、存储容量及工作可靠性明显提高	多用户、实时、数据库、软件产品; 软件技术的发展不能满足需要, 出现了软件危机
第三阶段	软件工程阶段	20 世纪 70 年代至 80 年代中期	程序、文档、数据	开发小组及大、中、小型软件开发机构	管理水平	向超高速、大容量、微型化发展	分布式系统、嵌入式软件出现; 软件危机并未摆脱
第四阶段	现代软件工程阶段	20 世纪 80 年代末至现在	程序、文档、数据、Web 页、方法、规则	中间件技术、网络技术、构件技术、管理技术的应用使跨平台、跨区域开发成为可能, 超大型软件工厂出现	管理水平	网络化发展迅速, 大容量新型高速存储器问世, 运算速度达每秒几万亿次至万万亿次的巨型计算机不断出现	强大的桌面系统、面向对象技术、专家系统、人工神经网络、并行计算、客户/服务器环境、面向 Agent 软件工程、网构软件技术、面向自适应和自组织系统的软件工程、面向组织的软件工程、大数据、云计算、“互联网+”等; 软件危机仍然存在

从表 1-3 可以看出, 在 20 世纪 60 年代由于软件的发展不能满足需求出现了软件危机, 所谓软件危机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, 这些问题几乎存在于所有的软件中。

软件危机主要表现为以下 6 个方面。

(1) 软件开发成本和进度估计不准确

据统计, 软件项目约有 40% 延期或超预算, 20% 不得不取消, 真正成功的软件项目案例不足 40%。实际成本比估计成本高出一个数量级, 实际进度比预期进度拖延几个月, 甚至几年的现象经常发生。例如, 微软公司的 Word 3.0 原定于 1986 年 7 月推出, 但是直到 1987 年 2 月才问世, 但问世后很快就发现有很多错误, 据说有 700 多处错误, 有的错误甚至会破坏数据, 摧毁程序, 微软不得不花费 100 万美元在两个月内为用户免费升级。

(2) 软件需求分析不充分, 开发出的系统与实际需求有差距

软件开发人员和用户之间的信息交流往往很不充分, 在没有确切了解问题的情况下, 或对用户需求还很模糊的情况下就开始编写程序, 导致开发出的软件产品与实际需求有差距, 甚至不符合用户的实际需要。在进行软件需求分析时, 不但要考虑功能需求, 还要考虑质量需求和约束, 既要有产品需求, 还要有市场需求和组件需求, 如果缺少必要的需求, 或得到错误的需求, 或不断变更的需求, 都会最终导致系统失败。例如, 哈维兰彗星 1 号客机, 由于没有考虑到方形窗口的承压能力而经常坠毁; 塔科马海峡大桥在需求和设计模型中没有考虑风力因素而被大风吹垮; Ariane 5 型火箭不恰当地复用了需求, 导致在第一次试航中坠毁。据克里斯托夫·埃伯特 (Christof Ebert) (如图 1-6 所示) 所著《需求工程: 实践者之路》一书中给出的数据, 87% 的项目失败是由于需求工程做得不够而引起的。

(3) 软件投入使用后经常出故障

一个著名的例子是美国 IBM 公司的 OS/360, 这是第一个功能较强的多道程序操作系统, 参加这项开发工作的有 IBM 公司美国国内的 11 个单位, 欧洲的 6 个单位, 共计 1000 多人, 耗费了 5000 人年的工作量, 但结果不尽人意, 它的每个版本都是在前一版本的基础上找出 1000 个程序错误而修正的结果, 项目总负责人费雷德里克·布鲁克斯 (Frederick

P.Brooks Jr., 见图 1-7) 在他所著的《人月神话》(The Mythical Man-Month) 一书中生动地描述了开发中遇到的困难和问题, 提出了很多令人深思的观点。



图 1-6 克里斯托夫·埃伯特



图 1-7 弗雷德里克·布鲁克斯

#### (4) 软件难以维护

软件的维护不像硬件那样通过进行零部件的修复、更换或系统整体升级就能做到, 修改了发现的错误后可能还会导致新的错误, 更换后的软件也不一定比旧软件好。例如, 美国第四大药品批发商福克斯迈尔医药公司在 1994 年决定用 C/S 系统和 SAP 软件代替老化的 Unisys 大型系统, 该公司每天要填写成千上万份药房订单, 每个药房又订购上千种药品, 总量达 50 万种, 与系统升级和改造相关的费用就花费了 3800 万美元。但在系统投入实际使用中却发现 SAP 软件每天只能处理几千条数据, 在短短几个星期内, 由于发出错误订单这一项, 就使公司遭受了 1550 万美元的损失, 最后迈基森公司为其支付了 8000 万美元现金, 才使其子公司没被破产, 总公司的股票也从 26 元跌到 3 元, 使公司蒙受了巨大损失。

很多程序中的错误很难改正, 让一个软件系统适应新的硬件环境, 或在原有程序中增加一些新功能都非常困难, 有时为了纠正软件的 1 个错误的工作量甚至可能要超出原来软件开发的总工作量。

#### (5) 缺乏文档资料

在软件开发过程中应该有一套与之同步生成的文档资料, 它们是软件开发过程的“里程碑”, 若缺乏文档资料或事后再补都会给开发及维护带来困难。在计算机诞生的早期, 软件规模比较小, 应用领域狭窄, 文档的作用并不十分突出, 但是随着计算机应用领域的扩大及软件技术的发展, 特别是软件工程的出现, 软件文档的地位和作用也越来越重要, 软件文档在软件产品开发过程中起到了重要的桥梁作用。例如, 20 世纪 70 年代初, 汽车中只有几个控制器, 需求规格说明书加在一起也不过上百页, 而现代汽车中有 50 多个控制器, 需求规格说明书达到了 10 万页。目前, 软件文档已经成为项目管理的依据、软件人员进行技术交流的语言和依据、也是进行项目质量审查和评价的重要依据, 它们为用户和技术人员提供培训和维护的资料、为维护软件提供技术支持, 它们还是未来项目的一种资源, 良好的系统文档有助于把程序移植和转移到各种新的系统环境中。

(6) 软件在计算机总成本中所占比例逐年上升，维护费用增长迅速

在计算机系统诞生的早期，硬件的价格决定了系统的整体价格，但是随着硬件技术的快速发展及价格的不断下降，硬件与软件在整个计算机系统里的比重快速发生着变化，到 20 世纪 60 年代它们的成本比例就发生了颠倒，软件成本成为主体，例如，某发电厂花几百万元购买的数据采集与监控系统（简称 SCADA 系统），其中的 90% 都是软件费用。现在硬件已经成为软件的“包装”，软件的成本就决定了整个系统的价格。据美国军方给出的数据，在 20 世纪 60 年代的 F-4 战斗机中，由软件来完成的功能约为整体功能的 8%，到 90 年代，在 B-2 轰炸机中，由软件来完成的功能就占到整体功能的 65%，而在 21 世纪，在 F-22 战斗机中，由软件来完成的功能占到整体功能的 80%，由此可看出软件已居主导。目前，软件费用已经成为计算机系统的主要费用，其中维护费用又占去了大部分，有些软件机构 70% 的工作量都花在维护已有的软件上。

随着计算技术的发展，软件危机的表现形式也在发生变化，旧的危机克服了，还会出现新的危机。为了摆脱日益严重的软件危机，1968 年在德国小镇加米施（Garmisch）召开的北大西洋公约组织（简称 NATO）学术会议上，与会学者们首次提出了软件工程的观念，图 1-8 所示是当时的会场，这次会议是软件开发走向工程化的标志，从此软件技术的发展及软件工程的研究有了长足的进步，并起着越来越大的作用。目前，软件工程已经成为一门大的学科，在它下面又出现了很多新的学科，需要研究的未知领域和课题还很多，但软件危机并未完全解决。



图 1-8 1968 年在 NATO 学术会议上首次提出了软件工程的观念

## 2. 软件工程的定义

软件工程既是工程，又是一门学科，软件工程学科的内容很丰富，定义也是多种多样，最早的定义是弗里斯·鲍尔（Frith Bauer）在 NATO 学术会议上给出的，他指出软件工程是建立和使用一套合理的工程原则，以便经济地获得可靠的、可以在实际机器上高效运行的软件。

IEEE 给出的软件工程定义是：软件工程首先是将系统化的、规范的、可量化的方法应用于软件的开发、运行和维护，即将工程化方法应用于软件；其次是与上述有关方法的研究。

美国国家工程院院士、著名的软件工程专家巴利·玻姆（Barry W. Boehm）（如图 1-9 所

示)给出的定义:软件工程是现代科学技术知识在设计和构造计算机程序中的实际应用,其中包括管理在开发、运行和维护这些程序的过程中所必需的相关文档资料。

美国软件工程专家 Roger S. Pressman 把软件工程视作一种层次化技术,软件工程的任何活动都必须建立在高质量的基础之上,支持软件工程的根基就在于高质量,他的层次技术思想可以用图 1-10 来描述。



图 1-9 巴利·玻姆

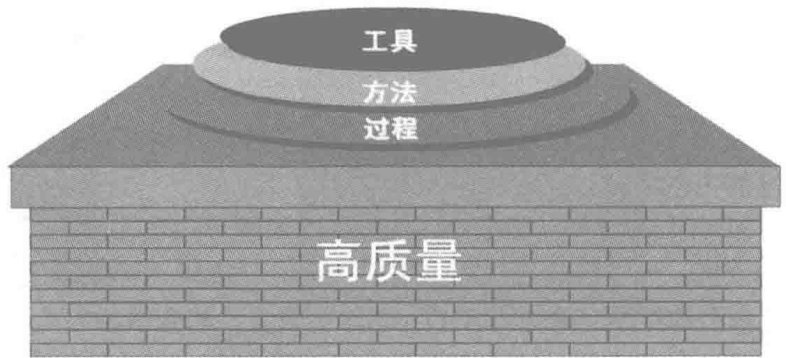


图 1-10 软件工程层次图

从图 1-10 可以看出,软件工程层次结构的基础是过程层,过程层定义了一个框架,构建该框架是有效实施软件工程技术必不可少的。软件过程构成了软件项目管理控制的基础,在软件过程中要建立工作环境以便于应用技术方法、提交工作产品(如模型、文档、数据、报告、表格等)、建立里程碑、保证质量及正确管理变更。

软件工程方法为构建软件提供技术上的解决方法,主要包括沟通、需求分析、设计建模、编程、测试和技术支持。软件工程技术方法依赖于一组基本原则,这些原则涵盖了软件工程所有技术领域,包括建模和其他描述性技术等。

软件工程工具为过程和方法提供自动化或半自动化的支持。这些工具可以集成起来,使得一个工具产生的信息可被另外一个工具使用,这样就建立了软件开发的支撑系统,称为计算机辅助软件工程。

北京大学王立福教授等给出的软件工程定义:软件工程是一类求解软件的工程。它应用计算机科学、数学及管理科学等原理,借鉴传统工程的原则、方法,创建软件以达到提高质量、降低成本的目的。其中,计算机科学、数学用于构造模型与算法,工程科学用于制定规范、设计泛型、评估成本及确定权衡,管理科学用于计划、资源、质量、成本等管理。软件工程是一门指导计算机软件开发和维护的工程学科。

从以上软件工程的定义可以看出,软件工程包含的内容很丰富,它涉及软件开发、管理、维护、质量保证等各个方面,它既有一般工程的特点,也有其特殊性,它不但和软件相关,也和其他学科相关,软件工程是一门多学科交叉的学科。

## 1.2 软件的生命周期

软件和任何有生命或无生命的事务一样,有它的生命周期(或称为生存周期),一个人