

计算机等级考试教程

(三级 B)

软件工程方法

全国高等学校计算机教育研究会
教材与课程建设委员会 组编

李大友 主编

机械工业出版社

计算机等级考试教程

(三级 B)

软件工程方法

全国高等学校计算机教育研究会
教材与课程建设委员会 组编

李大友 主编
屠立德 屠祁 编著



机械工业出版社

该书是根据国家教委计算机等级考试三级B(软件类专业)考试大纲及各省市考试大纲的要求编写的。该书内容的深度和广度完全符合大纲要求。

本书全面地介绍了软件工程方法的基本概念、基本原则和基本方法,以及软件的开发环境。内容包括:软件工程的基本概念、软件生存周期、可行性研究、需求分析、总体设计、编码方法、软件测试方法、软件维护、软件工程管理、软件开发环境的概念、软件工具、用户界面与用户界面开发系统和软件工程环境的数据库。该书具有内容先进、丰富。

本书可作为等级考试的软件工程教材,本科生、大专生“软件工程”和研究生“软件开发环境”教科书,也可作为软件开发人员的参考书。

图书在版编目(CIP)数据

计算机等级考试教程(三级B):软件工程方法/李大友主编.
—北京:机械工业出版社,1996.2
ISBN 7-111-04989-6

I. 计… II. 李… III. ①计算技术-基本知识-考试,等级-指导读物②软件工程-考试,等级-指导读物 IV. ①TP3
②TP311.5

中国版本图书馆CIP数据核字(95)第22605号

出版人:马九荣(北京市百万庄南街1号 邮政编码100037)
责任编辑:何文军 版式设计:张世琴 责任校对:韩晶
封面设计:郭景云 责任印制:卢子祥
三河永和印刷有限公司印刷·新华书店北京发行所发行
1996年2月第1版第1次印刷
787mm×1092mm¹/₁₆·18印张·429千字
0 001—6 000册
定价:26.00元

凡购本书,如有缺页、倒页、脱页、由本社发行部调换

《计算机等级考试教程》

编 委 会

主 编	李大友		
副主编	袁开榜	何 莉	陈瑞藻
编 委	(按姓氏笔划为序)		
	邓德祥	李芳芸	邵学才
	杨文龙	陈季琪	孟庆昌
	宗大华	姜秀芳	陶龙芳
	屠立德	葛本修	薛宗祥
秘 书	何文军		

《计算机等级考试教程》序言

当前,在世界范围内,一个以微电子技术、计算机技术和通信技术为先导的,以信息技术和信息产业为中心的信息革命方兴未艾。信息技术和信息产业的发展,对国民经济的发展、国家经济信息化起着举足轻重的作用,并已成为衡量一个国家发展水平的重要标志。因此,实现国家经济信息化,已成为世界各国所追求的共同目标。

为了使我国尽快实现国家经济信息化,赶上发达国家的水平,必须加速发展我国的信息技术和信息产业。其中最关键的环节就是人才的培养,尤其是计算机应用人才的培养。有了人才,才能迅速提高全社会的计算机应用水平,促进国家经济信息化水平的提高。因此,解决全民普及计算机知识,尽快提高全民族整体的计算机应用水平,已成为当务之急。各行各业、各层次人员,不论年龄与知识背景如何,都应掌握和应用计算机,解决其各自专业领域的计算机应用问题,为本职工作或专业服务,使其与国家经济信息化的需要相适应。

国家教委考试中心为适应这一形势发展的需要,使所培养的计算机应用人才的水平有一个公正的、客观的统一标准,推出了全国计算机等级考试。这一考试,根据应试者所具有的计算机应用能力水平的不同,划分为不同等级,分别进行考核。

全国计算机等级考试共分为四级六类,其内容范围如下:

一级分为 A、B 两类,均面向文字处理和数据库应用系统操作人员。

一级 A 类要求掌握计算机基础知识、微机系统基本组成、操作系统功能和使用、字表处理软件的功能和使用、数据库应用系统的基本概念和操作。

一级 B 类要求掌握计算机基础知识、微机系统基本组成、DOS 操作系统基本知识及操作、文字处理软件 WPS 和数据库语言 FoxBASE 的操作。

二级面向使用高级语言进行程序设计的人员。要求掌握计算机基础知识、操作系统的功能和使用、数据库的基本概念及应用和具有使用一种高级语言(C 语言、PASCAL 语言、FORTRAN 语言、BASIC 语言或数据库语言)进行程序设计的能力。

三级分为 A、B 两类。

三级 A 类面向测控领域的应用人员。要求掌握微机原理、汇编语言程序设计、微机接口技术、软件技术基础以及微机在测控领域的应用。

三级 B 类面向软件方面的应用人员。要求掌握计算机基础知识、数据结构与算法、操作系统、软件工程方法以及具有微机在管理信息系统或数值计算或计算机辅助设计方面的应用能力。

四级要求达到相当于大学计算机专业本科毕业生水平,具有计算机软件 and 硬件系统的设计开发能力。要求掌握计算机系统原理、计算机体系结构、计算机网络与通信、离散数学、数据结构与算法、操作系统、软件工程和数据库系统原理等方面的基础理论知识。

为推动全国计算机等级考试的健康发展,满足社会上对等级考试教材的迫切要求,全国

高等学校计算机教育研究会课程与教材建设委员会组织了高等院校多年从事计算机教育的第一线专家教授，编写了《计算机等级考试教程》系列教材，并得到机械工业出版社的大力支持与合作，使得这套教程能够及时与广大读者见面。

这套教程严格按照各级各类考试大纲的要求编写，内容深入浅出、图文并茂，每本书均附有习题，便于自学。

由于计算机技术是一门迅速发展的学科及作者水平所限，这套教程肯定会有很多不足之处，衷心希望得到社会各界和广大读者的批评指正。

主编 李大友

1995 年 11 月

前 言

自软件危机以来,软件工程技术的发展已有近 30 年的历史。当今软件工程技术已成为一切计算机应用领域最重要的基础性的学科,是当前世界上仅次于汽车工业而成为第二大产业——软件产业的主要技术支柱。因而各国政府和各界人士都十分关注软件工程技术在本国的普及、推广、应用和发展,并要求在中等学校中普及软件工程技术的教育。

软件工程是研究软件开发技术的一门学科,包括软件开发的技术方法、管理技术、工具和环境。与其密切相关的领域还涉及管理学、经济学、心理学、系统工程、艺术和法律等。软件工程是计算机领域中新技术的聚焦点,因此它又是一门发展迅速的新兴学科,新的技术方法和工具不断涌现,使之内容不断充实、丰富。

本书是作者近 10 余年来在本科生和大专生中讲授“软件工程”,以及在研究生中讲授“软件开发方法、环境和工具”的经验而编写的,编写目的有两个:

首先是应全国高等学校计算机教育研究会和教材与课程建设委员会之约,按等级考试大纲进行编写,作为应考者按考试大纲要求全面掌握有关软件工程的知识的教材。

其次是作为高等学校的本科生、大专生的软件工程和研究生的软件工程环境和工具课而编写的教科书。实际上,考试大纲与高等学校的教学大纲之间并无区别,不同的是考试大纲要求应考者综合地灵活地掌握软件工程知识(从历届考题中充分说明这一要求),而这也正是高等学校的教学要求,但从广度和深度上来说,两者是一致的。

本书具有以下特点:

1. 适合于重点讲授和自学的教学方式

本书语言通俗易懂、文字流畅简练,深入浅出,适合应考者自学。而对大学的教学来说,要求拓宽知识面,减少课程学时数而内容却应增加。因此,只有采取重点讲授,增加课后阅读内容来解决。

2. 实用性和实践性

本书包含有丰富的各类习题,以及历届我国和日本的程序员和高级程序员水平考试题,以检验自己了解程度和等级考试题目特性,培养解题技巧。

书中还包含有多个实际的软件工程环境和实用软件工具介绍,以及多个课程设计的实践题,例如开发 Windowsx.x 下的图形编辑工具。这些是软件工程技术人员在开发工作中很好的参考资料。

3. 内容先进性

本书介绍了最新的开发技术和软件开发环境。除传统的结构化技术外,还介绍面向对象、软件再用和快速原型的设计方法、软件开发环境和工具,研究了集成的概念和机制、以及环境信息库的发展情况。所有这些反映了当代软件工程技术最新的进展。

4. 学习的进化性

软件工程技术发展迅速,学习要不断进化。等级考试的参加者应学习全书的全部内容;大学本科学生应掌握第 1 篇全部内容,大专学生可不学习带有星号的章节。而研究生应以第 2 篇

全部内容为主，辅以第1篇中第1、3、4章中部分内容，如软件工程基本原则等。按软件工程专业发展，各类学生学习内容可以进化、提高。

本书第2篇主要介绍CASE和软件开发环境，按软件开发环境的逻辑模型和组成划分为五章。

本书有不当之处，敬请广大读者不吝赐教。

作者

计算机等级考试教程

目 录

1	一级	DOS 版	李大友等	编著
2	一级	WINDOWS 版	姜秀芳等	编著
3	二级	微机系统基础知识	李大友等	编著
4	二级	PASCAL 语言结构化程序设计	陶龙芳等	编著
5	二级	C 语言结构化程序设计	孟庆昌等	编著
6	二级	数据库语言程序设计 (FOXBASE+)	汪大菊等	编著
7	二级	FORTRAN 语言结构化程序设计	邓德祥等	编著
8	三级 A	微机原理与汇编语言程序设计	梁洪峻等	编著
9	三级 A	微机接口技术	李大友等	编著
10	三级 A	软件技术基础	李芳芸等	编著
11	三级 A	微机在测控领域的应用	陈季琪等	编著
12	三级 B	微机系统基础知识	薛宗祥等	编著
13	三级 B	数据结构	徐孝凯等	编著
14	三级 B	操作系统	匙彦斌等	编著
15	三级 B	软件工程方法	屠立德等	编著
16	三级 B	微机系统应用技术	李大友等	编著
17	四级	计算机系统组成及工作原理	葛本修等	编著
18	四级	计算机体系结构	金利杰等	编著
19	四级	网络与通信	王同胜等	编著
20	四级	离散数学	邵学才等	编著
21	四级	数据结构与算法	陈文博等	编著
22	四级	计算机操作系统	宗大华等	编著
23	四级	软件工程	姚淑珍等	编著
24	四级	数据库系统原理	李爱中等	编著

目 录

《计算机等级考试教程》序言

前 言

第 1 篇 软件工程原理

第 1 章 软件工程概论	3	3.4.3 数据词典的实现和使用	34
1.1 软件工程产生的背景与软件危机	3	3.5 数据处理的描述方法	35
1.1.1 软件开发技术的发展时期	3	3.5.1 数据处理的描述	35
1.1.2 软件危机	5	*3.5.2 结构化语言	36
1.1.3 软件工程	6	3.5.3 判定表	37
1.2 软件生存周期	7	*3.5.4 判定树	40
1.2.1 软件生存周期的概念	7	3.5.5 三种描述方法的比较	41
1.2.2 软件开发的瀑布模型	8	3.6 需求分析阶段的其它工作	42
1.3 软件质量评价	10	习题	43
习题	12	第 4 章 总体设计	45
第 2 章 可行性研究	14	4.1 引言	45
2.1 可行性研究的任务	14	4.2 软件工程学的基本原则	45
2.2 可行性研究的步骤	14	4.2.1 模块化	46
2.3 系统流程图	16	4.2.2 抽象	47
习题	18	4.2.3 信息隐藏	49
第 3 章 需求分析	19	4.2.4 信息局部化	51
3.1 需求分析的基本概念	19	4.2.5 其它原则	51
3.1.1 需求分析的任务	19	4.3 模块独立性的度量	51
3.1.2 需求分析步骤	20	4.3.1 模块独立性的概念	51
3.1.3 软件需求规格说明书	21	4.3.2 模块独立性的度量之一： 耦合性	52
3.2 结构化分析 (SA) 方法	22	4.3.3 模块独立性的度量之二： 内聚性	56
3.2.1 自顶向下逐层分解	22	*4.3.4 耦合、内聚与模块独立性关系	59
3.2.2 描述方式	22	4.4 软件结构的优化准则	60
3.2.3 SA 导出逻辑模型的步骤	23	4.5 结构设计中的图形工具	66
3.3 数据流图	23	4.5.1 层次图和 HIPO 图	66
3.3.1 数据流图的成分和对 系统的分解	23	4.5.2 结构图	67
3.3.2 怎样画数据流图	26	4.6 面向数据流的设计方法	70
3.4 数据词典	31	4.6.1 面向数据流设计方法的 基本概念	70
3.4.1 数据词典的内容	31	4.6.2 变换型数据流图到软件 结构的设计	74
3.4.2 数据的定义方法	33		

4.6.3 事务型数据流图到软件结构的设计	76	6.3.6 效率	124
*4.6.4 综合型数据流图到软件结构的设计	76	习题	125
*4.6.5 导出软件初始结构中的有关问题	78	第7章 测试	126
*4.6.6 设计优化	79	7.1 测试的基本概念	126
4.7 总体设计的其它工作	83	7.1.1 什么是软件测试	126
习题	84	7.1.2 测试的基本原则	127
第5章 详细设计	87	7.1.3 软件测试步骤及与开发阶段的关系	128
5.1 详细设计的概念	87	7.1.4 软件测试信息流	129
5.2 结构化程序设计(SP)方法	88	7.2 软件测试策略	130
5.3 详细设计的描述方法	91	7.2.1 静态测试方法与动态测试方法	130
5.3.1 程序流程图	92	7.2.2 白盒测试与黑盒测试	131
5.3.2 盒图(N—S图)	92	7.2.3 穷举测试	132
5.3.3 问题分析图(PAD)	94	7.2.4 人工测试	132
*5.3.4 把非结构化程序变换为结构化程序的方法	97	7.3 白盒测试技术	135
5.3.5 过程设计语言(PDL)	98	7.3.1 逻辑覆盖	135
5.3.6 模块开发卷宗	101	*7.3.2 最少测试用例数计算	139
5.4 Jackson 程序设计方法	102	*7.3.3 测试覆盖准则	141
5.4.1 Jackson 方法的设计步骤	102	*7.3.4 符号测试	141
5.4.2 Jackson 方法的三种基本结构和图式	102	7.4 黑盒测试技术	143
5.4.3 由数据结构的对应关系导出程序结构	104	7.4.1 等价类划分	143
5.4.4 列出和分配操作	106	7.4.2 边界值分析	147
5.4.5 用伪码表示程序	107	*7.4.3 因果图	148
5.4.6 结构冲突	109	7.4.4 错误推测和判定表驱动测试	150
*5.5 Warnier 程序设计方法	109	7.4.5 综合策略	150
*5.5.1 Warnier 图	109	7.5 单元测试	151
*5.5.2 Warnier 方法	110	7.6 集成测试	153
习题	112	7.6.1 自顶向下结合	154
第6章 编码	115	7.6.2 自底向上结合	155
6.1 引言	115	7.7 验收测试	156
6.2 程序语言的选择	115	7.7.1 验收测试标准	156
6.3 程序设计风格	118	7.7.2 软件配置复查	157
6.3.1 程序内部的文档	118	*7.7.3 Alpha (α) 和 Beta (β) 测试	157
6.3.2 变量命名与匈牙利命名规则	119	7.8 系统测试	158
6.3.3 数据说明	122	7.9 调试	158
6.3.4 语句构造	123	7.9.1 调试方法	158
6.3.5 输入/输出	124	*7.9.2 调试策略	159
		习题	161
		第8章 软件维护	163
		8.1 软件维护的概念	163

8.1.1 软件维护的定义	163
8.1.2 软件维护的生产特性	164
8.2 软件维护的困难和问题	165
8.2.1 结构化与非结构化的维护	165
8.2.2 软件维护的成本	166
8.3 软件维护的组织和实施	167
8.3.1 软件维护的组织	167
8.3.2 维护报告	168
8.3.3 软件维护的事件流	168
8.3.4 存档软件维护的记录	168
8.4 软件维护的副作用	169
8.4.1 修改代码的副作用	170
8.4.2 修改数据的副作用	170
8.4.3 文档的副作用	170
8.5 软件配置管理	171
8.5.1 软件配置管理的概念	171
8.5.2 软件配置项 (SCI)	172
8.5.3 软件配置管理的内容	172
8.5.4 软件配置管理的处理	172
8.5.5 软件文档管理	174
习题	176

第9章 软件工程管理	178
9.1 软件项目计划	178
9.1.1 软件项目计划的概念	178
9.1.2 软件项目计划内容	179
9.1.3 制定和实施软件工程规范	180
9.2 成本估算	181
9.2.1 COCOMO 模型	181
9.2.2 Putnam 估算模型	183
9.2.3 软件成本估算的其它方法	183
9.3 进度计划管理	184
9.3.1 软件开发任务的并行性	184
9.3.2 Gantt 图	185
9.3.3 工程网络图	186
9.3.4 估算进度	187
9.4 资源管理	188
9.4.1 人员资源	188
9.4.2 硬件资源	189
9.4.3 软件资源	189
9.5 人员组织	190
9.6 软件质量保证	190
习题	191

第2篇 计算机辅助软件工程——软件开发环境

第10章 软件开发环境的概念	195
10.1 软件开发环境的类型和发展	195
10.1.1 什么是软件开发环境	195
10.1.2 软件开发环境的发展	196
10.1.3 软件开发环境的类型	197
10.2 对软件开发环境的要求	202
10.3 软件开发环境的逻辑模型	203
10.3.1 逻辑模型	203
10.3.2 处理	204
10.3.3 方法	205
10.3.4 自动化	206
10.4 软件开发环境的构成和框架	207
10.4.1 软件开发环境的构成	207
10.4.2 软件工程环境的框架	209
习题	213
第11章 软件设计方法	214
11.1 引言	214
11.2 面向对象的设计方法	215
11.2.1 面向对象设计的概念	215
11.2.2 面向对象方法的设计步骤	217

11.2.3 面向对象方法的其它设计技术	218
11.3 面向可再用的软件设计方法	224
11.3.1 软件再用的基本概念	224
11.3.2 可再用设计方法的工作模式	226
11.3.3 可再用构件块的描述模式	226
11.4 原型方法	228
11.4.1 原型方法的概念	229
11.4.2 计算机辅助原型系统	230
习题	232

第12章 软件工具	233
12.1 软件工具的基本概念	233
12.1.1 什么是软件工具	233
12.1.2 当前软件工具发展的特点	234
12.1.3 软件工具的分类	236
12.1.4 软件工具的评价和选择	238
12.1.5 几种主要工具简介	241
12.2 工具间集成的概念	243
12.2.1 工具的集成性	243

12.2.2 集成性与灵活性	245	13.3.3 界面实现	259
12.3 基于程序设计系统的 结构设计工具 (SPD)	246	13.3.4 通信能力—— 对话控制层的主体	260
12.3.1 SPD 的基本概念	246	13.3.5 对应用的连接—— 应用界面模型	261
12.3.2 SPD 的成分	247	习题	263
习题	249	第 14 章 软件工程环境的数据库	264
第 13 章 用户界面与用户界面 开发系统 (UIMS)	250	14.1 软件工程环境信息库的作用 和对它的要求	264
13.1 图形用户界面 (GUI)	250	14.1.1 环境信息库的作用	264
13.1.1 图形用户界面的概念	250	14.1.2 对环境信息库的要求	265
13.1.2 图形用户界面的标准化	251	14.2 传统数据库的不足	266
13.2 用户界面开发系统 (UIMS)	254	14.2.1 数据库的非传统应用	266
13.2.1 用户界面开发系统的结构 模型	254	14.2.2 传统数据库对非传统 应用的不适应性	268
13.2.2 UIMS 的任务和组成	255	14.3 可扩充的数据库系统	270
13.2.3 UIMS 的描述方法	255	14.3.1 面向应用的扩充	270
13.3 UIMS 的实例—— Hyper News 系统	258	14.3.2 面向系统的扩充	270
13.3.1 Hyper News 简介	258	14.4 可扩充的 DBMS 实例	271
13.3.2 Hyper News 的对象类层次模型 与设计模型	258	习题	274
		参考文献	274

第 1 篇

软件工程原理

第1章 软件工程概论

软件工程是近30年来新出现,并得到迅速发展的计算机软件的重要学科。它是与各国软件产业的兴起和发展紧密相联的。众所周知,在欧美等经济发达国家,软件产业在国民经济中地位已成为仅次于汽车工业的重要的经济支柱,所以软件产业的主要技术基础——软件工程必然地受到各国的全力关注,大力地应用、推广和开发软件工程技术。这样就推动和促进了近10年来软件工程在世界上蓬勃发展的局面。

软件工程的名称是在1968年,由北大西洋公约组织,在原联邦德国的加尔密斯召开的软件可靠性国际会议上第一次提出。那么当前这样受重视的软件工程技术,为何在计算机技术出现了20多年以后才发展起来,软件工程产生的背景是什么?

1.1 软件工程产生的背景与软件危机

1.1.1 软件开发技术的发展时期

软件工程是研究软件生产和软件开发、维护的技术,所以软件工程的诞生是与软件开发技术的发展密切相关的。

从本世纪40年代计算机出现后,软件开发技术一直伴随着计算机硬件技术的发展而发展,到目前,软件开发技术大致经历了三个发展时期:

1. 个体生产时期(又称程序时期)

这个阶段是计算机应用的早期时代,它大约从40年代后期到60年代初。这个阶段的应用特点之一是,应用软件的开发、使用者和维护者,往往是同一个人。因为在计算机应用的初期并不存在专门的软件开发人员。要想用计算机来解决问题,往往由使用者自己来编写程序。而且还要由本人设法使程序在机器上运行操作,并且负责使这个程序来解决自己的问题。如果程序在运行过程中发生错误,也要由自己来修改。所以程序的开发者,使用者和运行维护者全是同一个人。这种个体特性是这个阶段的突出特点,它使得程序完全不需要考虑是否易于被其他人读和理解的要求,也不需要向他人就程序中难理解的部分作任何解释和交待。因为当时的程序不是给其他人读的,而是为解决自己的问题而要由计算机执行的,所以这个阶段只有程序的概念,没有软件的概念,所谓的软件也就是程序本身。所以有人把这个阶段称为程序时期。这一个时期的另一个重要特性是非常重视编制程序的技巧以求提高程序的运行效率,这是由于当时硬件成本很高,如能使程序编制得占用较少的主存和使用较少的CPU机时,是十分有意义的。所以当时为了追求程序的效率,而完全不顾程序的结构是否清晰,是否易于被他人读和理解,这是特点之二。最后这个时期的程序是为每个具体的应用和具体的机器设备而专门设计的。缺乏通用性。这就使程序更没有被他人阅读的必要。

2. 软件作坊时期(又称“软件=程序+说明”时期)

这个时期大约是从60年代初到70年代初,此时由于计算机硬件技术的发展使得CPU速度和主存容量都有了很大提高,这就带动了应用软件系统的发展,使得程序规模不断增大,这样庞大的程序规模(例如需几十人年)终于使得单个用户无力开发,于是产生了软件作坊的

生产方式。其特点是多人分工、共同协作来编制一个程序。可以想像,这种软件开发方式带来的一个直接要求是程序开发者之间必须相互通信,相互理解有关程序是如何工作的,并且相关程序间要有一致的约束规范。这是这个时期的特点之一。其次随着程序规模和程序复杂性增加,程序已不再是随硬件奉送的附属品,在计算机应用愈来愈普及和深入的情况下,随着各种通用软件的大量出现,人们开发软件已不是为开发者个人使用,而是为了广泛的销售。于是软件变成了产品,并形成了新兴的软件产业。软件的商品化是这个时期的第二个特点,它为软件开发技术和软件本身带来了极为深远影响和巨大的发展动力。最后由于程序规模和复杂性增加,同时也便于共同的程序开发者之间的通信和协作交流,所以要求程序必须要有注解和说明,以便他人易于读懂和理解程序。因而这个时期软件的概念发展为“软件=程序+说明”,有人也把这个时期称为“软件=程序+说明”时期。

3. 软件工程时期

这个时期是从 70 年代至今。近 50 年来计算机硬件,无论是 CPU 速度还是存储容量都有突飞猛进的发展,CPU 速度提高了 10^{10} 倍,主机成本每 2~3 年降低一半,硬件的性能价格比每 10 年提高一个数量级。随着计算机硬件的发展,计算机软件的研制和应用也发生巨大的变化。但是与硬件相比,软件的发展不是那么顺利,在 60 年代末期终于暴露出了用软件作坊方式生产的软件,无论从质量上到数量上都难以满足要求,有些软件是根本无法维护的。“软件危机”开始出现了,1968 年北大西洋公约组织的计算机科学家在原联邦德国召开的会议上讨论如何摆脱这一困境时,正式提出并使用“软件工程”这个术语,一门新兴学科从此诞生了。

自软件工程诞生至今的 20 多年中,无论是在软件工程的方法、原理,还是技术等方面均有很大发展。粗略地说,软件工程的发展也已经历了两个时期。

(1) 软件工程方法时期 自 70 年代初到 80 年代中的软件工程初期,出现了不少指导软件开发的方法,如软件生存周期、软件开发模型、结构化分析和结构化设计 (SA-SD)、技术审查和管理复审等。这些方法是有效的,但是实施这些方法的许多工作是复杂而很费人工的(例如数据字典的编制和维护)。并且这些方法的实施(无论是文档的编写,还是程序的编制)当时以手工为主,很少使用计算机辅助技术(自动工具)。因而这个时期实际上对软件开发的质量和生产率(数量)的改善并不突出。另外,这个时期逐渐地发展了计算机辅助软件工程 (CASE——Computer Aided Software Engineering) 技术,开发并使用了不少自动工具(例如高功能的编译程序;语法制导的编辑程序;代码生成工具;交互式编程环境和面向结构的正文和图形的 CASE 工具等)。但这个时期的工具往往是各自开发的,工具之间不具备兼容性,即一个工具的输出信息不能作为另一个工具的输入数据(往往要进行数据格式转换或其它加工要求)。因为工具间不具备集成性,所以这种工具使用的效果也是有限的。

(2) 集成的计算机辅助软件工程时期 (ICASE Integrated CASE)。这个时期从 80 年代中到现在,这个时期的特点是以使用集成的计算机辅助软件工程 (ICASE) 环境为特点。这种软件开发环境的软件工程信息库 (repository) 中包含有支持多种软件开发模型(如瀑布模型、快速原型模型、增量开发模型等)与多种软件开发方法学(如结构分析和结构设计 SA-SD 方法,面向数据结构的 Jackson 方法,面向对象的方法,基于软件再用的方法等),并支持软件生存周期全过程的众多的软件工具。这些软件工具之间是高度集成的,它们不但相互兼容,而且能自动控制工具的执行序列以协同地完成预定任务。毫无疑问,这个时期软件开发的质量和生产率得到大大的提高,开创了软件工程的新局面。