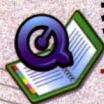


 高等教育规划教材

软件工程实用教程

周元哲 编著



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

软件工程实用教程

周元哲 编著



机械工业出版社

本书较为全面、系统地讲述了当前软件工程领域的理论和实践知识，涉及了最新的软件工程理论、标准、技术和工具，展望了软件工程的发展趋势。

本书内容包括软件工程学概述、软件可行性与项目计划、软件需求分析、软件设计、编程、测试、软件项目验收与维护、软件项目管理、软件过程、面向对象开发与 UML、软件工具与开发环境、文档、软件工程进展、软件工程案例和软件工程实验指导。

本书适合作为高等院校软件工程相关专业的教材或教学参考书，也可供计算机应用开发人员参考或用作全国软件技术资格与水平考试的培训资料。

本书配有授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ：2966938356，电话：010-88379739）。

图书在版编目（CIP）数据

软件工程实用教程 / 周元哲编著. —北京：机械工业出版社，2015.3
高等教育规划教材

ISBN 978-7-111-50887-8

I. ①软… II. ①周… 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（2015）第 162655 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：郝建伟 责任编辑：郝建伟

责任校对：张艳霞 责任印制：刘 岚

涿州市京南印刷厂印刷

2015 年 8 月第 1 版 · 第 1 次印刷

184mm×260mm · 13.75 印张 · 339 千字

0001—3000 册

标准书号：ISBN 978-7-111-50887-8

定价：36.00 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）88379833

机工官网：www.cmpbook.com

机工官博：weibo.com/cmp1952

读者购书热线：（010）88379649

教育服务网：www.cmpedu.com

封面防伪标均为盗版

金书网：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

软件工程是本科计算机相关专业及信息管理与信息系统专业的主干课程，也是电子信息工程专业的选修课程。它是一门理论与实践紧密联系的课程，直接关系到学生的理论分析能力和综合动手能力的培养。本书简要介绍软件开发生命周期的各个阶段的理论内容，重点介绍每阶段所需完成的图表以及如何采用相关的软件工具完成这些图表。

软件工程理论繁杂，让学生在实践中学理论知识，并用理论知识指导实践是本书的写作目的。本书培养学生按照软件工程的原理、方法、技术、标准和规范进行软件开发的能力，培养学生的合作意识和团队精神，从而提高学生软件开发的综合能力。

本书内容精练、文字简洁、结构合理、综合性强，明确定位面向初、中级读者，由“入门”起步，侧重“提高”。特别适合作为高等院校相关专业软件工程课程的教材或教学参考书，也可供计算机应用开发人员参考。

本书内容包括软件工程学概述、软件可行性与项目计划、软件需求分析、软件设计、编程、测试、软件项目验收与维护、软件项目管理、软件过程、面向对象开发与 UML、软件工具和开发环境、文档、软件工程新进展、软件工程案例和软件工程实验指导。

西安电子科技大学计算机学院鱼滨教授对本书的编写给予了大力的支持并提出了指导性意见。西安邮电大学计算机学院刘海、浙江大学城市学院周苏教授对本教材的大纲、写作风格等提出了很多宝贵意见。

本书在写作过程中参阅了大量中外专著、教材、论文、报告及网上资料，由于篇幅所限，未能一一列出。在此，向各位作者表示敬意和衷心的感谢。

由于作者水平有限，时间紧迫，本书难免有不足之处，诚恳期待读者的批评指正，以使本书日臻完善。

目 录

出版说明

前言

第1章 软件工程学概述	1
1.1 软件	1
1.1.1 软件发展历史	1
1.1.2 软件特点	1
1.1.3 软件分类	3
1.2 软件危机	3
1.2.1 软件缺陷案例	3
1.2.2 软件危机产生原因	4
1.3 软件工程研究范畴	5
1.3.1 软件开发方法	5
1.3.2 软件开发工具	6
1.3.3 软件开发过程	7
1.4 软件开发模型	8
1.4.1 瀑布模型	8
1.4.2 原型模型	10
1.4.3 增量模型	10
1.4.4 螺旋模型	11
1.4.5 喷泉模型	12
1.5 软件工程基本原理	13
1.6 软件产业	14
1.6.1 国内外软件产业的现状	14
1.6.2 对软件人才的要求	14
小结	15
习题	15
第2章 软件可行性与项目计划	16
2.1 可行性研究目的	16
2.2 可行性研究内容	16
2.2.1 分析现有系统	16
2.2.2 技术可行性	16

2.2.3 经济可行性	17
2.2.4 其他可行性	17
2.3 可行性研究步骤	18
2.4 项目计划	19
2.4.1 项目计划作用	19
2.4.2 项目计划种类	19
2.4.3 项目计划内容	20
2.4.4 项目计划目的	21
小结	21
习题	22
第3章 软件需求分析	23
3.1 需求分析内容	23
3.2 需求分析困难	24
3.3 需求分析步骤	24
3.4 相关图表	25
3.4.1 实体关系图	26
3.4.2 数据流图	27
3.4.3 数据字典	29
3.4.4 状态转换图	30
3.4.5 层次方框图	30
3.4.6 加工逻辑工具	31
小结	32
习题	32
第4章 软件设计	33
4.1 软件设计概述	33
4.1.1 数据结构设计	33
4.1.2 接口设计	34
4.1.3 过程设计	34
4.2 软件设计的基本原理	34

4.2.1 模块化设计	34	习题	65
4.2.2 耦合性和内聚性	35	第6章 测试	67
4.2.3 改进软件结构的指导 原则	37	6.1 测试历史演变	67
4.3 两种设计方法	38	6.2 测试目的和原则	67
4.3.1 面向数据流设计方法	38	6.3 测试模型	68
4.3.2 面向数据结构设计方法	39	6.3.1 V 模型	69
4.4 软件体系结构	41	6.3.2 W 模型	69
4.4.1 二层模式——C/S 体系 结构	41	6.3.3 H 模型	70
4.4.2 三层模式——MVC 模式	42	6.3.4 X 模型	70
4.5 人机界面设计	43	6.4 测试用例	71
4.5.1 用户特性	43	6.5 测试方法	71
4.5.2 人机界面设计原则	44	6.5.1 白盒测试	71
4.5.3 人机界面设计四个问题	46	6.5.2 黑盒测试	74
4.6 相关图表	47	6.6 测试生命周期	81
4.6.1 程序流程图	47	6.6.1 测试计划	81
4.6.2 N-S 图	48	6.6.2 测试分析	81
小结	49	6.6.3 测试设计	82
习题	50	6.6.4 测试执行	82
第5章 编程	51	6.6.5 评估测试	87
5.1 编程语言	51	6.7 自动测试技术	88
5.1.1 程序设计语言历史	51	6.7.1 自动化测试适用场合	88
5.1.2 结构化程序设计	52	6.7.2 测试工具原理	88
5.1.3 选择语言的标准	53	小结	89
5.2 高级程序设计语言	54	习题	90
5.2.1 面向过程程序设计语言	54	第7章 软件项目验收与维护	92
5.2.2 面向对象程序设计语言	55	7.1 软件项目验收	92
5.2.3 面向过程与面向对象的 区别	57	7.1.1 项目结束	92
5.3 编码规范	59	7.1.2 项目验收过程	92
5.3.1 编码风格	59	7.1.3 项目移交与清算	93
5.3.2 程序效率	60	7.1.4 项目评价	94
5.4 程序调试	63	7.2 软件维护	95
5.4.1 调试策略	63	7.2.1 软件维护种类	95
5.4.2 三种调试工具	63	7.2.2 软件维护过程	96
小结	65	7.2.3 软件维护副作用	98
		7.2.4 软件可维护性	98
		小结	98
		习题	99

第 8 章 软件项目管理	100	小结	132
8.1 项目管理与软件质量.....	100	习题	133
8.1.1 项目管理.....	100	第 10 章 面向对象开发与 UML	134
8.1.2 软件质量.....	101	10.1 面向对象开发过程.....	134
8.1.3 软件测试与质量保证.....	103	10.1.1 面向对象分析.....	134
8.2 成本管理.....	103	10.1.2 面向对象设计.....	134
8.2.1 代码行法.....	104	10.1.3 面向对象实现.....	135
8.2.2 功能点法.....	104	10.2 统一建模语言.....	137
8.2.3 构造型成本模型法.....	106	10.2.1 UML 简介.....	137
8.3 进度管理.....	107	10.2.2 UML 模型图概述.....	138
8.3.1 图示方法.....	107	10.2.3 用例图、时序图、状态图 举例说明.....	138
8.3.2 活动历时估计.....	108	10.3 UML 在三种模型中的 应用.....	140
8.4 人力资源管理.....	110	10.3.1 对象模型.....	140
8.4.1 项目人员角色.....	110	10.3.2 动态模型.....	144
8.4.2 项目组织结构.....	110	10.3.3 功能模型.....	148
8.5 风险管理.....	112	小结	149
8.5.1 风险类型.....	112	习题	149
8.5.2 风险管理主要活动.....	113	第 11 章 软件工具与开发环境	151
8.6 配置管理.....	114	11.1 软件工具与开发环境.....	151
8.6.1 相关概念.....	114	11.2 软件工具分类.....	151
8.6.2 配置管理流程.....	115	11.3 通用项目管理工具.....	152
小结	116	11.4 分析与设计工具.....	153
习题	117	11.4.1 Project.....	153
第 9 章 软件过程	118	11.4.2 GanttProject.....	155
9.1 过程改进.....	118	11.4.3 Excel.....	156
9.2 ISO 9000 系列.....	118	11.4.4 Visio.....	156
9.2.1 ISO 9000-3 介绍.....	119	11.4.5 PowerDesigner.....	156
9.2.2 ISO/IEC 9126 标准.....	120	11.5 软件测试工具.....	158
9.3 CMM/PSP/TSP.....	120	11.5.1 测试管理软件 TestDirector.....	158
9.3.1 软件能力成熟度模型.....	120	11.5.2 功能测试工具 WinRunner.....	158
9.3.2 个体软件过程.....	122	11.5.3 性能测试工具 LoadRunner.....	158
9.3.3 群组软件过程.....	123	11.5.4 单元测试工具 JUnit.....	158
9.4 ISO/IEC 15504 过程评估.....	124		
9.5 其他软件过程.....	125		
9.5.1 Rational 统一过程.....	126		
9.5.2 敏捷过程.....	129		
9.5.3 微软解决方案框架.....	130		

11.6	缺陷跟踪工具	163	13.3	净室软件工程	180
11.6.1	Rational ClearQuest	164	13.3.1	净室软件工程历史	180
11.6.2	JIRA	164	13.3.2	净室软件工程内容	180
11.6.3	Bugzilla	165	13.4	软件再工程	182
11.7	软件配置工具	165	13.4.1	设计恢复	182
11.7.1	Subversion (SVN)	165	13.4.2	逆向工程	182
11.7.2	VSS	165	13.4.3	正向工程	183
11.7.3	ClearCase	166	13.4.4	重构	183
11.7.4	CVS	166	13.5	软件复用	183
11.8	UML 工具	166	13.5.1	软件复用分类	184
11.8.1	Rational Rose	166	13.5.2	软件复用技术	184
11.8.2	ArgoUML	167	13.6	面向方面的编程	185
小结		169	13.6.1	AOP 概述	185
习题		169	13.6.2	AOP 术语	186
第 12 章	文档	170	13.6.3	AOP 语言	187
12.1	文档作用	170	13.6.4	AOP 研究方向	187
12.2	文档编制原则	170	13.7	多 Agent 软件体系结构	187
12.3	文档分类	171	13.7.1	Agent 概述	187
12.3.1	可行性研究报告	172	13.7.2	面向对象与面向代理	188
12.3.2	项目开发计划	172	13.7.3	MAS 系统	189
12.3.3	软件需求说明书	172	13.7.4	MAS 优点	189
12.3.4	概要设计说明书	173	13.8	网格动态体系结构	190
12.3.5	详细设计说明书	173	13.8.1	基本概念	190
12.3.6	测试计划	174	13.8.2	网格的定义	190
12.3.7	测试分析报告	174	13.8.3	网格的技术特征	191
12.3.8	用户手册	174	13.8.4	网格的体系结构	191
12.3.9	项目开发总结报告	174	小结		193
小结		174	习题		193
习题		174	第 14 章	软件工程案例	194
第 13 章	软件工程进展	175	14.1	面向过程开发案例——学生宿舍管理系统	194
13.1	软件外包	175	14.1.1	需求分析	195
13.1.1	外包分类	175	14.1.2	软件设计	199
13.1.2	外包项目监理	176	14.1.3	系统实现	201
13.2	开源软件	176	14.2	面向对象开发案例——广告管理系统	202
13.2.1	开源软件文档	177			
13.2.2	开源软件协议	177			
13.2.3	开源软件简介	178			

14.2.1	需求分析	202	15.2.1	工资管理系统需求	207
14.2.2	软件设计	202	15.2.2	学校课程安排系统需求	207
14.2.3	系统实现	205	15.2.3	学校教材订购系统需求	208
小结		205	15.2.4	实验室设备管理系统需求	208
习题		205	15.2.5	机票预订系统需求	209
第 15 章	软件工程实验指导	206	小结		209
15.1	实验内容	206	习题		209
15.1.1	实验目的	206	参考文献		210
15.1.2	实验要求	206			
15.1.3	实验组织形式	206			
15.2	实验题目	207			

第1章 软件工程学概述

软件工程学主要研究如何以较小的代价获得高质量的软件。软件固有特点导致软件危机。而软件工程就是为了解决软件危机而产生的。本章就软件工程的内容，如软件开发方法、工具和开发过程进行了说明，介绍了几种软件开发过程模型，如瀑布模型、原型模型、增量模型和螺旋模型等，最后介绍了软件工程基本原理以及软件产业的发展情况。

1.1 软件

本节主要介绍软件的发展历史、软件的特点及软件的分类。

1.1.1 软件发展历史

自第一台计算机诞生以来，软件需求迅速增长。软件的发展演变大体上经历了程序、软件、软件产品等几个阶段，具体阶段如下所示。

1. 程序设计时代（1946~1956年）

在这一时期，软件的生产主要是依靠个体手工劳动的生产方式。程序设计者使用机器语言、汇编语言作为工具；开发程序的方法主要是追求编程技巧和程序运行效率。在程序设计中还没有注意其他辅助作用，因此所设计的程序难读、难懂、难修改。这个时期软件开发的特征是只有程序、程序设计概念，不重视程序设计方法。

2. 程序系统时代（1956~1968年）

随着计算机的应用领域不断扩大，软件的需求也不断增长，由于软件处理的问题域扩大而使程序变得复杂，设计者不得不由个体手工劳动转变成小集团合作，形成作坊式生产方式的小集团合作生产的程序系统时代。生产工具是高级语言。开发方法仍旧靠个人技巧。由于大的程序需要合作开发，在程序设计中开始提出结构化方法。

3. 软件工程时代（1968年至今）

北大西洋公约组织（NATO）在1968年举办了首次软件工程学术会议，讨论了软件危机的问题，并在会中提出“软件工程”术语，确定软件的生产方式采用工程的概念和方法进行开发。

1.1.2 软件特点

软件作为计算机系统中与硬件相互依存的一部分，具有以下内容。

- 1) 在运行中能提供所希望的功能与性能的程序。
- 2) 使程序能够正确运行的数据及其结构。
- 3) 描述软件研制过程和方法所用的文档。

Frederick Brooks 在《没有银弹》（发表于 IFIP 第 10 届世界计算大会会议录）中总结了软件的以下特点：

1) 软件是逻辑实体，而不是物理实体。软件可以记录在纸面上，保存在计算机的存储器中，也可以保留在磁盘、磁带等介质上，但无法看到软件的形态，它不具有空间的形体特征。

2) 软件不会“磨损”，但会退化。一般情况下，有形的硬件产品在使用过程中总会磨损。在使用初期，往往磨损比较严重，而经过了一段时间进入到相对稳定期，随着元器件老化等原因，整个硬件最后寿命快要到了。硬件的磨损与时间之间的关系如图 1-1 所示，具有“浴缸曲线”的形状。

软件不是有形的产品，因此也就不存在所谓的“磨损”问题，软件的故障曲线与时间的关系如图 1-2 所示。在软件的运行初期，由于未知的错误会使程序有较高的故障率，然而当修正了这些错误并且没有引入新的错误后，软件将进入到平稳运行期。软件尽管不会“磨损”，但会退化，因为软件在其生命周期中会经历多次修改，每次修改都会引入新的错误，而对这些错误又要进行新的修改，使得软件的故障曲线呈现出锯齿形。

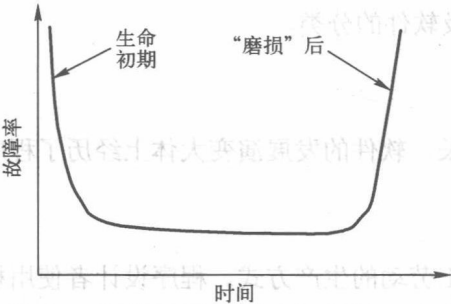


图 1-1 硬件故障率曲线

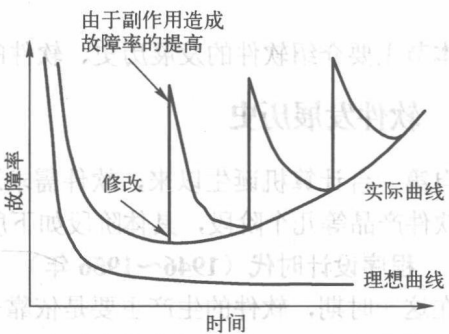


图 1-2 软件故障率曲线

3) 开发过程复杂。现代软件的体系结构越来越复杂，规模越来越庞大，涉及的学科也越来越多，导致软件开发过程异常复杂，软件的开发成本也越来越高。

软件项目具有如下特点：

1) 知识密集型，技术含量高。软件项目是知识密集型项目，技术性很强，需要大量高强度的脑力劳动。项目工作十分细致、复杂和容易出错。软件项目不需要使用大量的物质资源，而主要是使用人力资源，因此人员的因素极为重要，项目团队成员的结构、技能、责任心和团队精神对软件项目的成功与否有着决定性的影响。

2) 涉及多个专业领域，多种技术综合应用。软件项目属于典型的跨学科合作项目，例如开发大型管理信息系统就需要项目成员具有行业的业务知识、数据库技术、程序设计技术和信息安全技术等多专业领域知识。

3) 项目范围和目标的灵活性。随着项目的进展，客户需求可能会发生变化，从而导致项目范围和目标的变化。衡量软件是否成功的重要标准就是用户满意度，这个标准在软件开发前很难精确地、完整地表达出来。

4) 风险大。由于技术的高度复杂性和需求等因素的不确定性，软件项目风险控制难度较大，项目的成功率较低。

5) 客户化程度高。软件开发商往往要根据客户的具体要求提供独特的解决方案，即使

有现成的解决方案，也通常需要进行一定的客户化工作。

6) 过程管理的重要性。“质量产生于过程”，需要对整个项目过程进行严格和科学的管理，监控软件开发的中间结果。

软件项目管理强调阶段性和过程性，在实施项目管理的过程中，应该把项目划分为便于管理的一系列阶段，并在不同的阶段进行相应的项目管理活动。

1.1.3 软件分类

当今社会，软件应用渗透到社会的各行各业，从不同的角度对软件进行分类，如表 1-1 所示。

表 1-1 软件的分类

分类方法	对应类别	典型应用与特征
按功能分类	(1) 系统软件	与计算机硬件的接口并为其其他程序服务，如操作系统、驱动程序等 用于开发软件的工具性软件，如开发平台、数据库管理系统、种种工具软件等 为解决某一领域而开发的软件，如商业软件、嵌入式软件、个人计算机软件、Web 软件、人工智能等
	(2) 支撑软件	
	(3) 应用软件	
按版权分类	(1) 商业软件	版权受法律保护、经授权方可使用且必须购买的软件 与商业软件类似，但可以“先尝后买”，其获取途径主要是通过因特网 无须支付许可证费用便可得到和使用的软件，发行渠道类似于共享软件 没有版权，任何人均可以使用而且可以获得源代码的软件
	(2) 共享软件	
	(3) 开源软件	
	(4) 公有领域软件	
按工作方式分类	(1) 实时软件	用于即时处理实时发生的事件的软件，如控制、订票系统等 多个联机用户同时使用计算机的软件 能够实现人机通信的软件 多个作业或多批数据一次运行，顺序处理的软件
	(2) 分时软件	
	(3) 交互式软件	
	(4) 批处理软件	
按销售方式分类	(1) 定制软件	受某个特定的客户委托，在合同的约束下而开发的软件 由软件开发机构开发可以为众多用户服务的，并直接提供给市场的软件
	(2) 产品软件	

1.2 软件危机

软件危机是指落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现系列严重问题的现象。本节介绍了软件危机的表现形式以及软件危机产生的原因等相关内容。

1.2.1 软件缺陷案例

1963 年，由于用 FORTRAN 程序设计语言编写的飞行控制软件中的循环语句 DO 5 I=1, 3 误写为 DO 5 I=1.3，DO 语句少了一个逗号，导致美国首次金星探测飞行失败，造成价值约 1000 多万美元的损失。

1979 年，新西兰航空公司的一架客机因计算机控制的自动飞行系统发生故障而撞在阿尔卑斯山上，机上 257 名乘客全部遇难。

1983 年，美国科罗拉多河泛滥，但由于计算机对天气形势预测有误，水库未能及时泄洪，以致造成严重的经济损失和人员伤亡。

1992 年 10 月 26 日, 伦敦救护中心的计算机辅助系统刚启动就崩溃, 导致当时全世界最大的每天要接运五千多病人的救护机构全部瘫痪。

1994 年, 美国迪斯尼公司的《狮子王》游戏软件由于与一部分用户操作系统不兼容, 导致游戏崩溃。愤怒的家长们将迪斯尼告上了法庭, 迪斯也因此蒙受了巨大的经济和名誉损失。

1996 年 6 月 4 日, 耗资 80 亿美元的欧洲航空航天局发射的阿里亚娜 501 火箭, 发射升空后爆炸。原因是主发动机打火顺序开始 37s 后, 制导信息由于惯性制导系统的软件出现规格和设计错误而完全遗失。

2000 年, 著名的“千年虫”问题。计算机早期的硬件资源珍贵, 程序员为节约内存资源和硬盘空间, 在存储日期数据时, 只保留年份的后 2 位, 如“1980”被存储为“80”。当 2000 年到来的时候, 问题出现了。如银行存款的软件在计算利息时, 本应该用日期“2000 年 1 月 1 日”减去当时存款的日期。但是, 由于计算机无法分清“00”是指“2000”还是“1000”, 结果用“1000 年 1 月 1 日”减去存款时的日期, 存款年数就变为负数, 导致存款者反要付给银行利息。

.....

1.2.2 软件危机产生原因

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题, 主要是指如何开发软件, 怎样满足对软件日益增长的需求, 如何维护数量不断膨胀的现有软件。软件危机主要有如下表现形式。

1) 对软件开发成本和进度的估计很不准确。由于缺乏软件开发经验和软件开发数据的积累, 使得开发工作的计划很难制定。主观盲目制定的计划, 执行起来和实际情况有很大差距, 使得开发经费一再突破。由于对工作量和开发难度估计不足, 进度计划无法按时完成, 开发时间一再拖延。

2) 开发的软件产品不能完全满足用户要求, 用户对已完成的软件系统不满意的现象常常发生。有时, 软件开发人员与用户未能就需求进行十分明确的认定, 导致开发工作开始后, 一些问题不能及时解决, 致使开发的软件产品不能完全满足用户要求。

3) 开发的软件可靠性差。由于在开发过程中, 没有确保软件质量的体系和措施, 在软件测试时, 又没有严格的、充分的、完全的测试, 导致提交给用户的软件质量可靠性较差。

4) 软件通常没有适当的文档。开发过程无完整、规范的文档, 发现问题后进行杂乱无章地修改, 导致即使发现错误也难以更正。

5) 软件的可维护性差。由于开发过程没有统一的、公认规范, 软件开发人员各行其是, 导致程序的错误非常难以改正。

软件危机产生的具体原因如下所示。

1) 用户对软件的需求描述不清楚, 可能有遗漏、二义性, 甚至在软件的开发过程中, 用户还可能会提出修改软件功能、界面、支撑环境等方面的要求。

2) 开发人员对需求理解与用户的本来愿望有差异, 由于软件开发人员精通“技术”, 而用户只熟悉相关“业务”, 两者的交流不在一个平台上, 这种差异必然导致软件产品与用户需求的不一致。

3) 软件开发是团体的智力创造, 软件项目需要组织一定的人力共同完成, 各类人员的

信息交流不及时，不准确，有时还会产生误解。

4) 缺乏有力的方法学和工具方面的支持，过分依靠程序设计人员在软件开发过程中的技巧和创造性。

1.3 软件工程研究范畴

为了克服软件危机，1968 年在联邦德国召开的国际会议上提出了“软件工程”的概念。IEEE 对软件工程的定义如下：软件工程是将系统化的、严格约束的、量化的方法应用于软件的开发、运行和维护，即将工程化应用于软件。软件工程是一门综合性的交叉学科，它涉及哲学、计算机科学、工程科学、管理科学、数学及应用领域知识。

软件工程研究的内容主要集中在软件的开发技术与管理两大方面。开发技术包括软件的开发模型、开发过程、开发方法、工具与环境等；管理技术包括人员组织、项目计划、标准与配置、成本估算、质量评价等。

为了提高软件生产力改善软件质量和降低软件成本，软件工程从“过程”“方法”和“工具”三方面进行控制。软件工程的层次结构如图 1-3 所示，“工具”可以对软件工程的过程与方法提供自动或半自动的支持。在适当的软件工具辅助下，开发人员可以既快又好地做好软件开发工作，这些工具被称为 CASE（计算机辅助软件工程）工具。“方法”提供了实现软件过程的技术，它涉及一系列的任务：需求分析、开发模型、设计、编码、测试和支持等。“过程”是指必须为软件开发流程规定一系列的关键过程域，以此作为软件项目管理控制的基础，通过人员组织管理、项目计划管理、质量管理等环节来保证软件开发按时保质保量地完成。



图 1-3 软件工程的层次

1.3.1 软件开发方法

软件开发方法就是指导研制软件的某种标准规程，指导人们“什么时候做什么以及怎么做”。由于软件研发过程毕竟是比较复杂的，软件方法通常规定了明确的工作步骤，具体的描述方式以及确定的标准。

1. 明确的工作步骤

软件项目开发需要考虑和解决许多问题，必须将诸多问题安排好先后顺序，每一步集中精力解决一个问题。软件开发方法提供了处理问题的基本步骤，其中包括每一步的目的、产生的工作结果、需具备的条件以及要注意的问题等。

2. 具体的描述方式

工程化生产必须强调文档化，也就是说每一个人必须将自己的工作结果以一定的书面形式记录下来，以保证软件开发人员之间有效的交流，也有利于维护工作的顺利进行。软件方法规定了描述软件产品的格式，这包括每一步应该产生什么文档、文档中记录些什么内容、采用哪些图形和符号等。

3. 确定的评价标准

对于同一个问题，解决方案往往是不唯一的，应具体方案具体评价。

在长期的软件开发实践中，人们总结出如下几类常用的软件开发方法。

(1) SASD 方法

1978 年，E.Yourdon 和 L.L.Constantine 提出了结构化方法，即 SASD 方法，也可称为面向功能的软件开发方法或面向数据流的软件开发方法。作为 20 世纪 80 年代使用最广泛的软件开发方法，它首先采用结构化分析（SA）对软件进行需求分析，然后采用结构化设计（SD）方法进行设计，最后才进行结构化编程（SP）。

(2) 面向对象的软件开发方法

面向对象技术是软件技术的一次革命，在软件开发史上具有里程碑的意义。面向对象的软件开发方法包括 OOA（面向对象分析）、OOD（面向对象设计）和 OOP（面向对象编程），它以对象建模为基础，用于描述系统的数据结构；用动态模型描述系统的控制结构；用功能模型描述系统的功能。在需求分析、可维护性和可靠性等软件开发的关键环节有所提高，从而较好地保证了软件的质量。

1.3.2 软件开发工具

软件开发工具是指能支持软件生存周期中某一阶段（如系统定义、需求分析、设计、编码、测试或维护等）的需要而使用的软件系统。在软件开发工具的基础上形成了软件开发环境。软件开发环境是面向软件整个生存周期，为支持各个阶段的需要，在基本硬件和宿主软件的基础上使用的一组软件系统。1985 年第八届国际软件工程会议提出的关于“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。”软件开发工具和软件开发环境都是软件工程的重要支柱，对于提高软件生产力，改进软件质量以及适应计算机技术的迅速发展有着越来越大的作用。

通常，软件开发环境一般由环境数据库、接口软件和工具组等构成。

1) 环境数据库。作为软件开发环境的核心，环境数据库是指被研制软件在其生存周期中所必须的信息和软件研制工具的有关信息等。在环境数据库中，可以分别标识的信息组称为对象，作为一组相关的版本而存在的一组对象称为版本组，把一个项目中各个不同的“对象组”组合起来，就可以构成各种不同的软件配置。

2) 接口软件。接口软件包括系统与用户的接口、子系统和子系统之间的接口，开发环境要求所有的接口都具有统一调用方式。

3) 工具组。软件方法和工具是软件开发的两大支柱，软件方法作为设计软件工具的基础，提出了明确的工作步骤和标准的文档格式，而软件工具的实现又将促进软件方法的推广和发展。

软件开发环境中的工具彼此有交互作用。工具组中的各个工具被设计成由一些基本功能成分组成。这些成分可以组合，供用户选用，并且可通过环境数据库进行通信。

1) 语言工具。它一般提供编译（或解释）程序、连接装配程序、调试程序、静态及动态分析程序等一系列语言支撑工具。也有的系统支持非过程语言的运行。

2) 质量保证工具。目前大量使用的仍是静态、动态测试技术以及各种形式的评审技术。与形式方法相联系的程序证明和验证技术已为许多专家所重视。

3) 需求分析及设计工具。主要由文本编辑工具、图形工具及一致性检验工具等支持。

4) 配置管理工具。对于不同用户或不同的硬件配置，一个系统往往需不同的软件配

置；即使同一软件也有不同的版本；加上软件产品的修改需要进行严格的管理等原因，软件的配置管理已成为软件生产管理的重要课题。

1.3.3 软件开发过程

任何有生命的动物、植物与人都有一个生存周期，例如人的生存周期就是：胎儿——婴儿——幼儿——儿童——少年——青年——中年——老年——死亡。而没有生命的事物或实体如计算机、汽车等也有一个生存周期，这种生存周期实际上就是使用寿命。软件生存周期是指从提出软件产品开始，直到该软件产品被淘汰的全过程。软件生存周期要求软件的开发必须分阶段进行，前一个阶段任务的完成是下一个阶段任务的前提和基础，而后一个阶段通常是将前一个阶段提出的方案进一步具体化。每一个阶段的开始与结束都有严格的标准，在每一个阶段结束之前都要接受严格的技术与管理评审。

软件生存周期大体分为三个时期：软件定义、软件开发、软件支持，每个时期又可以分为若干个阶段，如表 1-2 所示。

表 1-2 软件周期

生存阶段	周期序号	周期名称	生存阶段	周期序号	周期名称
软件定义	1	问题定义 可行性分析	软件开发	6	测试
	2	需求分析		7	软件发布或 安装与验收
软件开发	3	概要设计	软件支持	8	软件使用
	4	详细设计		9	维护或退役
	5	编码			

1. 软件定义阶段

软件项目或产品分为定制软件和非定制软件两种。定制软件是软件开发者与固定的客户签订软件开发合同，由软件公司负责该项目的开发；非定制软件则是由软件公司通过市场调研，并在人力、设备、风险抵御、资金与时间等方面都具备开发该产品的能力，从而决定立项开发。

软件定义作为软件生存周期的第一阶段，主要解决如下三个问题：

- 1) 问题的定义。对于定制软件，首先要根据用户所提出的书面材料（设计要求或招标文件），研究用户的基本要求，确定需要解决的问题；而对于非定制软件则要研究软件的基本应用场合与功能、用户群等。通过对问题的研究应该得到关于软件的问题性质、工程目标与基本规模等。
- 2) 可行性分析。可行性分析是为前一阶段提出的问题寻求一种到数种在技术上可行且在经济上有较高效益的解决方案，最主要的是对系统进行成本/效益分析。如果是定制软件，要决定是否参加投标或竞争；如果是非定制软件，则要决定是否进行开发。
- 3) 立项或签订合同。当进行了全面的可行性分析后，就可以立项或与用户签订正式的软件开发合同。

2. 软件开发阶段

开发阶段一般经过四个步骤：需求分析、设计、编码与测试、安装与验收。