



新世纪高等学校规划教材·软件工程系列

软件工程实用教程

任传成◎主 编

李丽 孟芸 徐硕博◎副主编



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社



新世纪高等学校规划教材·软件工程系列

软件工程实用教程

任传成◎主 编

李丽 孟芸 徐硕博◎副主编



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

图书在版编目 (CIP) 数据

软件工程实用教程 / 任传成主编. —北京: 北京师范大学出版社, 2017.9

新世纪高等学校规划教材·软件工程系列

ISBN 978-7-303-22811-9

I. ①软… II. ①任… 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 (2017) 第 215914 号

营 销 中 心 电 话 010-62978190 62979006

北师大出版社科技与经管分社网 www.jswsbook.com

电 子 信 箱 jswsbook@163.com

出版发行: 北京师范大学出版社 www.bnup.com

北京市海淀区新街口外大街 19 号

邮政编码: 100875

印 刷: 北京玺诚印务有限公司

经 销: 全国新华书店

开 本: 787 mm×1092 mm 1/16

印 张: 19.5

字 数: 446 千字

版 次: 2017 年 9 月第 1 版

印 次: 2017 年 9 月第 1 次印刷

定 价: 38.90 元

策划编辑: 赵洛育

责任编辑: 赵洛育

美术编辑: 刘 超

装帧设计: 刘 超

责任校对: 王 云

责任印制: 赵非非

版权所有 侵权必究

反盗版、反侵权举报电话: 010-62978190

北京读者服务部电话: 010-62979006-8021

外埠邮购电话: 010-62978190

本书如有印装质量问题, 请与印制管理部联系调换。

印制管理部电话: 010-62979006-8006

内 容 简 介

本书较为全面地介绍了软件工程的基本概念、过程模型、结构化方法和面向对象开发方法。本书共分9章，概述了软件工程基本原理、各种生命周期的过程模型，介绍了软件开发各阶段的任务、过程和方法，全面、系统、清晰地介绍了结构化方法、统一建模语言和面向对象方法。

本书强调软件工程的理论与实践相结合，含有一定量的实例和习题，可作为计算机科学与技术、软件工程、网络工程相关专业的教材，也可以作为软件开发人员的参考书。

前 言

在提出了“软件工程”概念后，软件工程作为一门学科迅速发展起来。它以提高软件质量，降低开发成本为目的，将系统化的、规范的、可度量的工程化方法应用于软件应用系统的开发、运行和维护的过程中，使经过实践检验而证明是正确的管理方法和当前最好的开发技术结合了起来，使软件产业成为信息时代最大的产业之一。

软件工程是应用型高等院校计算机科学与技术 and 软件工程学科各专业的一门重要的专业基础课程，本书在介绍软件工程的基本理论和传统软件工程开发及管理技术的基础上，重点介绍 UML 统一建模语言和面向对象的分析与设计方法，旨在帮助读者具有一定的应用软件开发能力。

本书的内容丰富，结构严谨，涵盖了软件工程基本理论、结构化分析与设计、面向对象分析与设计及软件测试等有关软件工程的主要知识单元，重要的知识点与相应实例相结合，讲解由浅入深，既保证知识点的科学性和先进性，又体现知识点的实际应用性，适合作为应用型高校计算机科学与技术、软件工程、网络工程等相关专业本科生软件工程课程的教材，也可以作为软件开发人员的参考书。

全书共 9 章。第 1 章概括软件工程的产生原因及基本原理，介绍软件生命周期和常见的过程模型，讨论软件工程师职业道德规范与要求；第 2 章介绍软件项目可行性研究，主要包括可行性研究的目的、任务、步骤、成本与效益分析、人力资源管理和项目计划等内容；第 3 章介绍软件项目需求分析，包括需求分析的原则、任务、方法、过程和需求过程管理，以及结构化构建需求分析模型等内容；第 4 章介绍结构化分析与设计方法，包括结构化分析与设计的由来、软件设计的基本原理、面向数据流的设计方法、软件总体与详细设计等内容；第 5 章介绍 UML 统一建模语言和面向对象方法，包括面向对象方法的基本概念、特征、建模方法、UML 的图分类和基本原理等内容；第 6 和 7 章介绍面向对象分析与设计方法，包括分析阶段建立目标系统的用例模型、静态模型和行为模型，设计阶段构建目标系统的软件体系结构、系统的分解与设计、对象的设计与优化等内容；第 8 章介绍软件实现，包括程序语言择取、程序设计风格，软件测试基本概念、原则、基本方法、测试策略，面向对象的测试和 Web 应用软件的测试等内容；第 9 章介绍软件维护，包括软件维护的概念、产生的原因、维护类型、维护过程、可维护性和软件再工程技术等内容。

本书由任传成（山东德州学院）主编，第 1 章由孟芸（河南理工万方科技学院）编写，第 2、3 章由李丽（山东德州学院）编写，第 4 章由徐硕博（山东交通学院）编写，第 5 章由胡凯

和韩金姝（山东德州学院）编写，第6、7章由鲁燕和任传成（山东德州学院）编写，第8章由于瑞玲和孙志卓（山东德州学院）编写，第9章由宋秀芹（山东德州学院）编写。全书由任传成统稿并审定。

由于作者水平有限，书中难免存在疏漏之处，恳请读者批评指正。作者电子邮箱：rench@dzu.edu.cn。

编 者
2017年7月

目 录

第 1 章 软件工程概述	1
1.1 软件的概述	1
1.1.1 软件的发展历程	1
1.1.2 软件的分类	2
1.1.3 软件的特点	3
1.2 软件危机	4
1.2.1 软件危机的定义及表现	4
1.2.2 产生的原因及解决途径	4
1.3 软件工程概述	5
1.3.1 软件工程的定义	5
1.3.2 研究内容	5
1.3.3 基本原理	7
1.4 软件生命周期	8
1.5 软件过程模型	9
1.6 软件工程师职业道德规范与要求	16
本章小结	16
习题	17
第 2 章 软件项目可行性研究	18
2.1 可行性研究的目的和任务	18
2.2 可行性研究的步骤	19
2.3 可行性分析报告	21
2.4 成本和效益分析	22
2.5 软件项目人力资源管理	23
2.6 软件项目的计划	26
2.6.1 软件工作范围	26
2.6.2 资源	26

2.6.3 进度安排	27
本章小结	30
习题	30
第3章 软件项目需求分析	31
3.1 需求分析概述	31
3.1.1 需求分析的特点	31
3.1.2 需求分析的原则	32
3.1.3 需求分析的任务	32
3.1.4 需求分析的方法	33
3.2 需求分析的过程	35
3.2.1 获取用户需求	36
3.2.2 建立的逻辑模型	37
3.2.3 编写需求文档	38
3.2.4 文档评审	38
3.3 结构化需求分析模型	40
3.3.1 数据流图	40
3.3.2 数据字典	50
3.3.3 加工逻辑描述方法	53
3.3.4 实体关系模型	56
3.4 需求过程管理	59
本章小结	61
习题	62
第4章 结构化分析与设计	63
4.1 结构化分析与设计概述	63
4.1.1 结构化分析与设计的由来	63
4.1.2 总体设计的概述	64
4.1.3 详细设计的概述	66
4.2 软件设计的基本原理	66
4.2.1 抽象与逐步求精	67
4.2.2 模块和模块化	67
4.2.3 信息隐藏和局部化	68
4.2.4 模块独立性	68
4.2.5 复用性设计	73
4.2.6 软件结构设计的优化原则	74
4.3 面向数据流的设计方法	77
4.3.1 软件结构的图形表示	77
4.3.2 数据流的类型	79
4.3.3 设计过程	80
4.3.4 变换分析	81

4.3.5 事务分析	85
4.3.6 软件结构的优化	86
4.4 总体设计说明书与评审	89
4.4.1 总体设计说明书	89
4.4.2 总体设计评审	89
4.5 软件的详细设计	90
4.5.1 详细设计的任务	90
4.5.2 详细设计的原则	91
4.5.3 详细设计的方法	91
4.5.4 详细设计的工具	92
4.6 详细设计说明书与评审	100
4.6.1 详细设计说明书	100
4.6.2 详细设计评审	100
本章小结	101
习题	101
第5章 面向对象方法与UML	103
5.1 面向对象方法概述	103
5.1.1 面向对象方法简介	103
5.1.2 面向对象方法的发展史及现状简介	104
5.1.3 面向对象方法的基本概念和特征	104
5.2 面向对象的建模方法	106
5.2.1 Booch 方法	106
5.2.2 OMT 方法	107
5.2.3 OOSE 方法	107
5.2.4 Coad-Yourdon 方法	107
5.3 统一建模语言概述	108
5.3.1 UML 的发展历程	108
5.3.2 UML 的应用	109
5.3.3 UML 的构成	109
5.4 UML 的图	111
5.4.1 用例图	111
5.4.2 类图和对象图	119
5.4.3 包图	126
5.4.4 活动图	129
5.4.5 状态图	133
5.4.6 顺序图	137
5.4.7 通信图	143
5.4.8 定时图	146
5.4.9 交互概览图	148

5.4.10 组合结构图	149
5.4.11 组件图	151
5.4.12 部署图	154
本章小结	157
习题	158
第 6 章 面向对象的分析	161
6.1 概述	161
6.2 建立用例模型	162
6.2.1 建立用例模型的过程	162
6.2.2 使用 EA 创建用例模型	165
6.3 建立静态模型	168
6.3.1 划分子系统	169
6.3.2 确定类	169
6.3.3 识别实体类、边界类和控制类	169
6.3.4 确定关系	171
6.3.5 确定属性	172
6.3.6 确定操作	174
6.3.7 使用 EA 建立类图	175
6.4 建立行为模型	177
6.4.1 顺序图	177
6.4.2 通信图	180
6.4.3 状态图	181
6.4.4 活动图	182
6.5 实例分析	184
6.5.1 自动提款机系统的用例模型	185
6.5.2 自动提款机系统的静态模型	187
6.5.3 自动提款机系统的行为模型	189
本章小结	191
习题	191
第 7 章 面向对象的软件设计	193
7.1 面向对象设计的概述	193
7.1.1 面向对象设计的过程	193
7.1.2 面向对象设计的准则	195
7.2 软件体系结构设计	196
7.2.1 概述	196
7.2.2 类别与重要性	197
7.2.3 分布式软件体系结构	197
7.2.4 分布式对象体系结构	201
7.2.5 分布式实现机制	201

7.2.6 体系结构框架	203
7.3 系统分解与设计	206
7.3.1 子系统设计	206
7.3.2 组织系统的两种方案	207
7.3.3 设计问题域子系统	208
7.3.4 设计人机交互子系统	211
7.3.5 设计任务管理子系统	212
7.3.6 设计数据管理子系统	213
7.4 对象设计和优化	217
7.4.1 识别设计类	218
7.4.2 设计类的属性	218
7.4.3 设计类的服务	219
7.4.4 精化和优化设计类	220
7.5 网上购物系统的架构设计	222
7.5.1 网上购物系统的需求描述	222
7.5.2 网上购物系统的用例图	223
7.5.3 网上购物系统的3层架构	223
7.5.4 网上购物系统的类/对象设计	224
本章小结	226
习题	226
第8章 软件实现	227
8.1 程序设计语言	227
8.1.1 程序设计语言的特性	227
8.1.2 程序设计语言的种类	228
8.1.3 程序设计语言的基本元素	230
8.1.4 程序设计语言的选择	233
8.2 程序设计风格	235
8.2.1 源程序文档化	235
8.2.2 数据说明	239
8.2.3 语句构造说明	239
8.2.4 输入/输出	241
8.3 软件测试概述	241
8.3.1 软件错误、缺陷和故障	242
8.3.2 软件测试的概念	244
8.3.3 软件测试的原则	244
8.3.4 软件测试的方法	245
8.3.5 软件测试与软件开发各阶段的关系	246
8.3.6 软件测试文档	247
8.4 软件测试的基本方法	248

8.4.1	白盒测试	248
8.4.2	墨盒测试	256
8.5	软件测试策略	262
8.5.1	单元测试	262
8.5.2	集成测试	265
8.5.3	确认测试	269
8.5.4	系统测试	270
8.5.5	终止测试	272
8.6	面向对象的测试	273
8.6.1	面向对象的特性影响测试	273
8.6.2	面向对象测试模型	274
8.6.3	面向对象测试策略	276
8.7	Web 应用软件测试	278
8.7.1	Web 应用软件的基本结构	278
8.7.2	Web 应用软件测试的内容与目的	279
8.7.3	Web 应用软件的测试类型	279
8.7.4	Web 应用软件的测试策略	282
8.8	软件调试	283
8.8.1	软件调试的目的	283
8.8.2	软件调试的原则	283
8.8.3	软件调试的步骤	284
8.8.4	软件调试的方法	285
本章小结		285
习题		286
第9章	软件维护	288
9.1	软件维护概述	288
9.1.1	软件维护的原因	288
9.1.2	软件维护的类型及策略	288
9.1.3	软件维护的特点	290
9.1.4	软件维护的过程	290
9.2	软件可维护性	292
9.3	软件再工程技术	294
9.3.1	软件再工程	294
9.3.2	软件再工程的过程	294
本章小结		296
习题		296
参考文献		297

第 1 章 软件工程概述

随着计算机技术的迅猛发展，计算机已经渗透到人们工作和生活的各个领域，一方面软件数量急剧膨胀，软件需求日趋复杂，另一方面软件维护的难度越来越大，开发成本难以估计，失败的软件开发项目屡见不鲜。这种状况严重妨碍了计算机技术的进步，为了摆脱困扰，一门新兴学科——软件工程应运而生。

1.1 软件的概述

软件对于现代人而言并不陌生，但是很多人对软件的理解并不准确，认为软件就是程序，软件开发就是编写程序，这种错误观念依然存在。

软件是计算机系统中与硬件相依存的另一部分，它包括程序、数据及其相关文档。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文资料。

1.1.1 软件的发展历程

自从 1945 年世界上第一台电子数字计算机问世以来，计算机经历了电子管、晶体管、集成电路到超大规模集成电路几个发展阶段，同时软件也经历了四个不同的发展阶段，如图 1.1 所示。

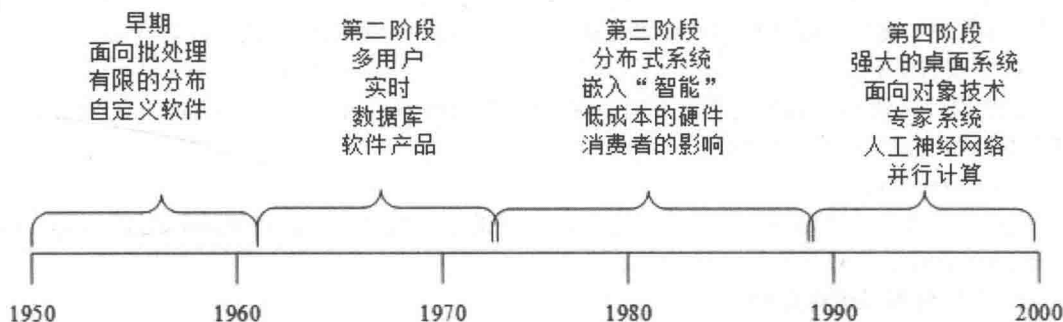


图 1.1 软件的发展

第一阶段指的是从 20 世纪 50 年代初到 60 年代初的 10 余年,是计算机系统开发的初期阶段。当时的软件开发几乎都是为某个具体应用专门编写的,开发人员往往是该软件的使用者。这种个体化的软件设计环境,使得软件设计成为人们头脑中的隐含过程,最终,软件呈现出来的只有程序清单,没有其他文档资料被保存下来。

除此之外,初期开发的计算机系统采用批处理技术,虽然这样能够提高计算机的使用效率,但是不利于程序的设计、调试、修改和维护。在这个阶段,人们认为计算机的主要用途是快速计算,软件编程相对简单,不存在任何系统化的开发方法,没有任何开发管理活动,程序的质量完全依赖于开发人员的个人编程技巧。

第二阶段是指从 20 世纪 60 年代中到 70 年代中的 10 余年。在这个阶段,很多用户系统引入了人机交互的概念,实时系统能够从多个数据源收集数据并加以分析和转换,在线存储设备的发展产生了第一代数据库管理系统。

软件产品和“软件作坊”的概念开始出现,开发人员开发软件不再像早期阶段那样只因个人研究需要而开发,而是为了用户更好地使用计算机,然而,“软件作坊”仍然沿用早期个体方式进行软件开发。程序运行时发现的错误必须设法更正;用户有了新需求时,必须相应地修改或添加程序;硬件或操作系统更新时,又可能要修改程序以适应新的环境。长此以往,软件的维护工作以惊人的比例耗费资源,更严重的是,程序设计的个体化和作坊化特性使软件最终无法维护,从而出现了早期的软件危机。

第三阶段是指从 20 世纪 70 年代中到 80 年代末的 10 多年。这个阶段的软件有不同的特点,如分布式系统极大地提高了计算机系统的复杂性,网络的发展对软件开发提出了更高的要求,特别是微处理器的出现和广泛应用,孕育了一系列的智能产品。硬件的发展速度已经超过了人们对软件的需求速度,随着硬件价格下降,软件的价格急剧上升,导致了软件危机的加剧,致使更多的科学家着手研究软件工程学的科学理论、方法和时限等一系列问题。软件开发技术的度量问题受到重视,最著名的有软件工程量估计模型 COCOMO、软件过程改进模型 CMM 等。

第四阶段是从 20 世纪 80 年代末开始。这个阶段是强大的桌面系统和计算机网络迅速发展的时期,计算机体系结构由中央主机控制方式变为客户机/服务器方式,专家系统和人工智能软件终于走出实验室进入了实际应用,虚拟现实和多媒体系统改变了与最终用户的通信方式,出现了并行计算和网络计算的研究,面向对象技术在许多领域迅速取代了传统软件开发方法。

1.1.2 软件的分类

经过多年的迅速发展,软件多种多样,功能也相差很大。在实际生活中,软件按不同的分类标准可分为不同类型的软件。

1. 按软件功能进行划分

(1) 系统软件

系统软件是计算机系统必不可少的组成部分,需要与硬件进行频繁的交互,负责各资源共享及进程管理、复杂数据结构的处理等工作,如操作系统、设备驱动程序等。

(2) 支撑软件

支撑软件是协助进行软件开发的工具软件,如 API (Application Programming Interface, 应用程序编程接口) 帮助文档等。

(3) 应用软件

应用软件在特定领域内开发,为特定目的服务的一类软件,如计算机辅助教学 (Computer Aided Instruction, CAI) 软件。

2. 按软件规模进行划分

根据开发软件过程中所需的人力、研制期限及完成软件程序包含的源代码行数,可将软件划分为 6 种不同规模的软件,如表 1.1 所示。

表 1.1 软件划分种类

类 别	所 需 人 力	研 制 期 限	产品规模 (源代码行数)
微型	1	1~4 周	500
小型	1	1~6 周	1000~2000
中型	2~5	1~2 年	5000~50 000
大型	5~20	2~3 年	50 000~500 000
超大型	100~1000	4~5 年	1 000 000
极大型	2000~5000	5~10 年	1 000 000~10 000 000

规模大、项目参与人员多、开发周期长的软件项目，其开发工作必须要有软件工程的理论知识作指导，而规模小、项目参与人员少、开发周期短的软件项目同样需要遵循软件工程的开发规范。

3. 按工作方式划分

(1) 实时处理软件

实时处理软件是必须满足严格时间约束条件的软件，一旦事件发生或数据产生，就需要对其及时处理，否则会影响系统的运行，甚至造成严重事故。

(2) 分时软件

允许多个联机用户同时使用的计算机软件，系统将处理时间划分成时间片，各联机用户轮流使用时间片，每个用户感觉自己独占计算机。

(3) 交互式软件

这类软件能实现人机通信。

(4) 批处理软件

批处理软件是把一组输入作业或一批数据以成批处理的方式一次运行，按顺序逐个处理的软件。

1.1.3 软件的特点

人们希望用类似工程的方法来解决软件开发中碰到的各种困难,但是与传统工业产品相比,软件具有以下一些典型特点。

(1) 软件是一种逻辑实体，不是具体的物理实体。这个特点使它与其他工业产品有着明显的差异。软件可以记录并存储在内存、磁盘或光盘等物理介质中，但却无法看到软件本身的形态，必须通过观察、分析、思考、判断，才能了解它的特征。

(2) 在软件的开发过程中没有明显的制造过程，一旦研制成功可以批量复制，所以要保证软件的质量，必须在软件开发方面下工夫。

(3) 在软件的运行和使用期间，没有机械磨损和老化问题，但存在退化问题。软件在投入使用后，为了适应硬件、环境及需求变化而需要多次对软件进行修改，但是每次修改会不可避免引入 Bug，导致软件失效率升高，从而引起退化。

(4) 软件的开发和运行往往受到计算机系统的限制，对计算机系统依赖性较强，因此需要软件具有可移植性。

(5) 软件的开发至今未摆脱手工作坊式的开发方式，生产效率低。

(6) 软件作为提高人类工作效率的逻辑产品，其本身是复杂的，而且以后会更加复杂。这主要表现在软件开发常常涉及其他领域的专业知识，势必对软件开发人员知识素养提出了更高的要求。

(7) 软件成本相当昂贵。软件开发需要投入大量、高强度的智力劳动，现在的软件开销已大大超过了硬件的开销。

(8) 软件开发牵扯到相当多的社会因素。许多软件开发和运行涉及机构、体制及管理方式等问题,甚至还会涉及人的观念和心里,这些人为因素,直接影响到项目开发的成败。

1.2 软件危机

1.2.1 软件危机的定义及表现

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严重问题。在 20 世纪 60 年代末到 70 年代初这段时间,软件危机一词在计算机界得以流传。事实上,软件危机几乎是从计算机诞生的那一天就存在了,只是到了 1968 年,北大西洋公约组织的计算机科学家在联邦德国召开的国际学术会议上才第一次提到了“软件危机”(Software Crisis)这个词。

一般来说,软件危机包含两方面问题:一方面如何开发软件,以满足不断增长、日趋复杂的需求,另一方面如何维护数量不断膨胀的已有软件产品。具体地说,软件危机主要有以下表现。

(1) 对软件开发成本和进度的估计常常不准确。这是因为软件缺乏“可见性”,一方面造成软件开发成本很难预估,开发经费不断追加,另一方面引起工作量估计不足,预先制定的进度计划无法遵循,开发工作完成期限一拖再拖,降低了开发组织的信誉。有些项目为了按期交付软件和节约成本而采取的一些权宜之计,然而这种方式会损害软件产品的质量。

(2) 用户对“已完成的”软件系统不满意的现象经常发生。一方面是用户在描述需求时表述不明确,或是在开发初期不能明确提出,而在软件开发过程中不断变更需求,同时双方未能及时沟通,造成软件开发后期出现大量问题;另一方面,在软件开发初期,软件开发人员不重视需求分析,对用户的需求只有一个大概的了解,在没有充分明确的前提下就开始匆忙着手编程,这种“闭门造车”的状态必然导致最终产品不符合用户实际需求。

(3) 软件产品的质量差强人意。软件项目即使能按预定计划完成,结果却差强人意。随着软件越来越多地用在国民经济和人类社会的高安全性、高可靠性系统中,软件的质量保证成为人们关注的焦点。

(4) 软件常常是不可维护的。投入使用的软件,总是出现一定数量的错误,例如不同运行环境,软件就会出现故障,这就需要维护。即使软件能正常使用,随着软硬件环境变化或用户提出新的需求,都需要对软件进行修改。由于在软件设计和开发过程中,没有严格遵循软件开发标准,各种工作随意性很大,没有完整记录系统状况的文档,给维护带来了极大的困难。更有甚者,原来的开发人员可能因为各种原因离开了开发组织,使得软件几乎不可维护。

(5) 软件通常没有适当的文档资料。计算机软件开发完成,不仅仅有程序,还应该有一套完整的文档资料。软件开发人员可以利用这些文档资料准确地交流信息,维护人员可以作为文档资料阅读从而掌握程序开发思路,以便快速查找问题所在,所以这些文档资料必不可少。

(6) 软件成本在计算机系统总成本中所占的比例逐年上升。由于微电子技术的进步和生产自动化程度不断提高,硬件成本逐年下降,而且软件开发需要大量的人力,软件成本随着通货膨胀及软件规模和数量的不断扩大而逐年上升。

1.2.2 产生的原因及解决途径

软件危机的出现暴露了软件产业在早期发展过程中存在的问题,为了克服软件危机,首先应该分析导致软件危机的原因。根据以上软件危机的种种表现和软件本身固有的逻辑结构,可以发现产生软件危机的原因包括以下几个方面。

(1) 软件开发需要用科学工程化的思想来组织和指导软件开发各个阶段。这种工程化的概念是很多软件开发人员所没有的,他们往往简单地认为软件开发就是程序设计。

(2) 没有完善的质量保证体系。没有严格的评审制度、科学的软件测试技术及质量维护

技术。

(3) 软件文档的重要性没有得到足够的重视。软件文档是软件开发团队成员之间交流和沟通的平台,也是软件开发项目管理的重要工具。缺乏必要的文档资料或文档资料不合格,将给软件开发和维护带来许多严重的困难和问题。

(4) 从事开发的专业人员对软件产业认识不充分,缺乏经验。

(5) 软件独有的特点给其开发和维护带来困难。软件的抽象性和复杂性使得软件在开发之前,很难对开发过程进行估计,再加上软件错误的隐蔽性和改正错误的复杂性,都使得软件开发和维护在客观上比较困难。

为了解决软件危机,人们逐渐认识了软件的特性及软件产品开发的内在规律,并尝试用工程化的思想去指导软件开发,于是软件工程诞生了。

1.3 软件工程概述

1.3.1 软件工程的定义

1968年10月在北大西洋公约组织召开的计算机科学会议上,Fritz Bauer首次提出了“软件工程”的概念,试图将工程化方法应用于软件开发,其对软件工程定义是:软件工程就是为了经济地获得可靠的且能在实际机器上有效地运行的软件,而建立和使用完善的工程原理。

随后,1993年《IEEE Standard Glossary of Software Engineering Terminology》给出了一个更全面的定义,软件工程是:①把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程,即将工程化应用到软件上;②研究①中所述方法的研究。

《计算机科学技术百科全书》中对软件工程的定义为: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、数学用于构建模型与算法,工程科学用于制定规范、设计规范、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。

2010年教育部高等学校软件工程专业教学指导分委员会在2010年制定的《高等学校软件工程本科专业规范》中对软件工程的定义为:应用计算机科学理论和技术及工程管理原则和方法,按预算和进度,实现满足用户要求的软件产品的定义、开发、发布和维护的工程或进行研究的学科。

软件工程定义很多,但主要思想都是强调软件工程不仅仅是软件开发技术的研究,还包括必要的组织管理措施。概括地说,软件工程是一门指导软件开发和维护的工程学科,采用工程的概念、原理、技术和方法来开发与维护软件,把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来,以经济地开发出高质量的软件并有效地维护它,这就是软件工程。

1.3.2 研究内容

软件工程方法学包括3个要素:方法、工具和过程。其中,方法是完成软件开发的各项任务的技术方法,回答怎样设计和开发软件的问题;工具是为了运用方法而提供的自动或半自动的软件工程支撑环境;过程是为了获得高质量的软件所需要完成的一系列任务框架,它规定了完成各项工作任务的工作步骤。

1. 方法

软件工程方法为软件设计和开发方面提供技术上的解决方法。这些方法包括沟通、需求分析、设计建模、编写代码、测试和维护支持。软件工程方法依赖于一组基本原则,这些原则覆盖了软件工程所有技术领域,包括建模活动和其他描述性技术。