



面向21世纪课程教材  
Textbook Series for 21st Century

# 软件工程

## (第3版)

齐治昌 谭庆平 宁洪



高等教育出版社  
HIGHER EDUCATION PRESS

面向 21 世纪课程教材  
普通高等教育“十一五”国家级规划教材  
国家精品课程主讲教材

# 软 件 工 程

Ruanjian Gongcheng

(第 3 版)

齐治昌 谭庆平 宁 洪



高等教育出版社·北京  
HIGHER EDUCATION PRESS BEIJING

## 内容简介

本书是在第2版的基础上修改而成的,阐述了信息时代软件、软件工程及软件工程教育的地位和作用,以及基于计算机的系统和业务过程建模方面的内容。书中分析了传统软件开发过程向统一过程 RUP 的进化,系统地介绍了 RUP 过程、UML 语言和面向对象的软件开发方法,以及软件开发的需求、设计、实现、测试、交付、维护、软件度量、软件项目管理和软件开发组织的过程改进等专题。本书还简化了结构化软件开发方法的相关内容,充实了目前常用的基于构件的软件开发、Web 软件工程等方面的内容。

本书强调理论与实践相结合,软件工程的技术、方法与工具相结合,软件项目的技术活动与管理活动相结合。书中含有丰富的例题、习题和参考文献。

本书可作为高等学校计算机专业或信息类相关专业高年级本科生教材,也可作为非计算机专业的研究生教材及软件开发人员的参考书。

## 图书在版编目(CIP)数据

软件工程/齐治昌,谭庆平,宁洪.—3版.—北京:高等教育出版社,2012.5

ISBN 978-7-04-032488-4

I. ①软… II. ①齐…②谭…③宁… 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(2012)第 014890 号

策划编辑 倪文慧 责任编辑 唐德凯 封面设计 于文燕 版式设计 范晓红  
插图绘制 尹莉 责任校对 杨凤玲 责任印制 张泽业

---

出版发行	高等教育出版社	网 址	<a href="http://www.hep.edu.cn">http://www.hep.edu.cn</a>
社 址	北京市西城区德外大街 4 号		<a href="http://www.hep.com.cn">http://www.hep.com.cn</a>
邮政编码	100120	网上订购	<a href="http://www.landaco.com">http://www.landaco.com</a>
印 刷	北京机工印刷厂		<a href="http://www.landaco.com.cn">http://www.landaco.com.cn</a>
开 本	787mm×1092mm 1/16	版 次	1997 年 4 月第 1 版
印 张	32		2012 年 5 月第 3 版
字 数	700 千字	印 次	2012 年 5 月第 1 次印刷
购书热线	010-58581118	定 价	42.00 元
咨询电话	400-810-0598		

---

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物料号 32488-00

## 第3版序言

当今世界已进入了信息社会和知识经济的新时代,软件产业飞速发展。为了给软件产业提供强大的人力资源和智力支持,软件工程教育出现了蓬勃发展的新局面。经过数年论证,2004年ACM、IEEE-CS联合推出了计算教程软件工程卷CCSE2004,与其配套的SWEBOK系统地阐明了软件工程学科的地位和知识体系,SEEK则给出了制定本科软件工程教学计划的指南。软件工程作为计算学科中的一个独立学科与计算机科学、计算机工程、信息系统、信息技术并列。同时,CC2004还将软件工程作为计算机科学、计算机工程、信息系统、信息技术学科的一个关键的知识领域,要求这些学科的学生应学习软件工程课程。我国的软件产业肩负着发展信息产业、对传统产业进行信息化改造的历史任务。为适应这一形势发展的需要,我国加快了软件人才培养的步伐,扩大软件人才培养规模,深化软件教学改革,创建软件学院,增设软件工程一级学科,系统地引进国外优秀的软件工程教材,扩大对外交流,短短几年取得了长足进步。

为适应软件工程教育发展的需要,本教材定位于高等学校本科生和没有学过软件工程课程的硕士研究生,是一门软件工程的综合性教材。大学本科阶段学过软件工程课程的研究生教材将另外编写。此外,为加强软件工程的课程实习,我们还编写了软件工程实践教程,供老师和学生选用。软件工程专业的教材需参照软件工程专业教育计划选用。

软件工程以计算机科学为基础,涉及数学、工程、管理、领域知识等诸多内容。良好的软件工程实践需要理论与实践相结合,需要技术和工具的支持,需要开发团队的共同努力。本书第3版在前两版的基础上,借鉴软件工程知识体系SWEBOK和SEEK的内容,针对我国高校本科软件工程教育的实际情况对内容进行选择和组织,主要内容包括软件与软件工程的概概念,软件开发的统一过程RUP和描述语言UML,软件的需求、设计、实现、验证和确认、维护和进化,软件度量、软件项目管理和软件组织的过程改进。

本书强调下列9个问题。

1. 软件中蕴涵的领域知识和经验。这是软件具有一定的功能和性能、能够为用户服务的基本保障。领域知识和经验必然涉及知识产权的处理,必然要求软件工程师学习必要的领域知识才能与领域专家有效沟通、合作。为配合领域知识的获取,在软件工程的统一过程RUP中增加了业务过程建模 workflow。

2. 软件生存周期的阶段划分与软件开发过程分解分开。这是近代软件工程的统一过程RUP与传统的软件开发瀑布模型的根本区别。RUP的软件生存周期展现了软件孕育、生长进化、交付生产、维护进化和退役的制品状态。RUP的软件过程包括:业务建模 workflow、需求 workflow、设计 workflow、实现 workflow、验证与确认(V&V) workflow、部署 workflow配置和变更管理 workflow、项

目管理 workflow 和软件开发环境 workflow。这些 workflow 按照 RUP 统一过程完成软件制品的进化。

3. 用统一建模描述语言 UML 描述 RUP 过程中的制品。用 UML 描述领域业务过程模型、软件需求模型、软件体系结构,支持软件的分析建模和体系结构设计。在此基础上,采用 Java 语言或 C++ 语言可全面、自然地实现面向对象的编程。

4. 验证与确认贯穿 RUP 过程的始终。通过走查、评审、测试等手段减少缺陷及缺陷的蔓延。

5. 变更管理和配置管理。软件开发过程中,由于种种原因,需求、设计、代码、文档等软件制品的变更不可避免,软件开发过程的软件配置管理、软件版本的增量发布等必须有方法和工具的支持才能保持软件制品的一致性和质量。

6. 瀑布模型、结构化软件开发方法、受限 C 语言在实时嵌入式系统中的应用。这些领域对软件的正确性、可靠性要求极高,必须强化软件的过程管理和控制,必须强化软件的测试和系统试验。

7. 软件的交付、现场安装、生产、维护和进化。这是软件发挥经济和社会效益的重要过程,也是用户在实践中检验软件的功能和性能、开发和服务质量的过程。软件工程组织和项目团队的所有研究、开发工作和努力都以此为出发点和归宿。

8. 软件工程的发展与硬件发展和社会需求的相互依赖关系。软件工程课程具有鲜明的时代特征,尽管本书篇幅和课时有限,还是介绍了 Web 软件工程、软件能力成熟度模型集成(CMMI)等方面的内容。

9. 系统的、规范的、可量化的软件开发、运行和维护。除重视软件开发的全部过程外,还特别重视软件和软件工程的度量,这有助于提高软件制品质量,有助于软件成本估算,帮助制定和实施有效的软件项目管理计划,帮助改进组织的软件开发过程。

针对上述问题的深入研究还需借助参考文献的帮助。本书列出的大部分参考文献都是 SWEBOK 推荐的,有广泛的影响。

为便于学生学习,系统地掌握软件开发的全过程,本书选用了两个有代表性的案例:“课程注册管理系统”和“家庭保安系统”。本书介绍的软件开发过程和方法始终结合这两个案例。

借再版机会,我们对书中的部分词汇的使用、疏漏、错误进行了修订。希望本版教材能够促进软件工程课程的教学内容改革,提高学生的软件开发能力和项目管理能力。

本书的第 1、15、16 章由齐治昌编写,第 2、3、4、5、7、8、9、14 章由谭庆平编写,第 6、10、11、12、13 章由宁洪编写。

在教学计划中,建议安排 60 学时,采用第 1 章至第 16 章的自然顺序讲授。课程实践应安排 20 学时以上。

复旦大学的钱乐秋教授认真地审阅了书稿,并提出许多宝贵的意见,指出了书稿中的许多疏漏,在此表示衷心的感谢。同时,向所有使用本书进行教学的老师和同学,以及对本书再版工作

提出建议和意见的专家和读者表示诚挚的谢意。

欢迎大家继续对本书给予支持并提出宝贵的意见。

作者

2011年11月

## 郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010)58581897 58582371 58581879

反盗版举报传真 (010)82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

# 目 录

<b>第 1 章 软件与软件工程</b> .....	1	1.7.2 基于计算机的系统的功能组件	43
1.1 软件的概念 .....	2	1.7.3 典型的基于计算机的系统	45
1.1.1 软件与软件的组成 .....	2	1.7.4 基于计算机的系统的研制	46
1.1.2 软件生存周期 .....	3	小结 .....	51
1.1.3 软件的特点 .....	3	习题 .....	52
1.1.4 几类常用的软件 .....	4	<b>第 2 章 UML 与 RUP 统一过程</b> .....	53
1.1.5 软件的质量 .....	7	2.1 案例说明 .....	53
1.1.6 软件的安全与保护 .....	9	2.2 面向对象的软件开发方法 .....	54
1.2 软件工程的概念 .....	10	2.2.1 面向对象的概念 .....	54
1.2.1 软件工程的定义 .....	10	2.2.2 面向对象方法的优势 .....	55
1.2.2 软件工程的发展 .....	12	2.3 UML 概述 .....	56
1.2.3 软件工程的目标和原则 .....	15	2.4 RUP 统一过程 .....	58
1.2.4 软件开发的主要方法 .....	17	2.4.1 RUP 软件过程框架 .....	58
1.3 软件过程模型 .....	18	2.4.2 RUP 的 5 个阶段 .....	60
1.3.1 瀑布模型 .....	19	2.4.3 RUP 的 9 个工作流 .....	62
1.3.2 增量过程模型 .....	22	小结 .....	66
1.3.3 原型建造模型 .....	23	习题 .....	66
1.3.4 螺旋模型 .....	24	<b>第 3 章 需求工程概论</b> .....	67
1.3.5 基于构件的过程模型 .....	25	3.1 软件需求的概念 .....	68
1.3.6 通用软件过程模型 .....	26	3.1.1 软件需求的分类 .....	68
1.4 敏捷软件开发原则和应用 .....	29	3.1.2 软件需求的质量要素 .....	68
1.4.1 敏捷软件开发原则 .....	29	3.2 需求工程的预备知识 .....	69
1.4.2 敏捷软件过程的极限编程实践 .....	30	3.2.1 与用户交流的技巧 .....	69
1.4.3 敏捷软件开发过程的应用 .....	31	3.2.2 需求调查的基本方法 .....	70
1.5 软件工程人的因素 .....	32	3.2.3 需求建模的基本方法 .....	72
1.5.1 软件工程教育 .....	32	3.3 需求工程的过程模型 .....	73
1.5.2 软件工程学科 .....	33	3.3.1 需求工程中的活动 .....	73
1.5.3 软件工程从业人员的职业道德 .....	35	3.3.2 迭代式的过程模型 .....	75
1.6 软件工具及软件开发环境 .....	35	3.3.3 过程模型的裁剪 .....	76
1.6.1 软件工具 .....	36	小结 .....	79
1.6.2 软件开发环境 .....	39	习题 .....	79
1.7 基于计算机的系统 .....	40	<b>第 4 章 需求获取</b> .....	83
1.7.1 基于计算机的系统的概念 .....	40	4.1 软件需求的初始表示 .....	83



4.1.1 用例 .....	83	5.4.2 设置分析类 .....	138
4.1.2 用例图 .....	84	5.4.3 构思分析类之间的协作关系 .....	141
4.1.3 用例的表示 .....	87	5.4.4 导出分析类图 .....	145
4.1.4 类图 .....	88	5.5 利用快速原型辅助需求分析 .....	151
4.1.5 活动图 .....	92	5.5.1 分析与规划 .....	152
4.2 需求获取的过程模型 .....	95	5.5.2 设计与实现 .....	153
4.3 定义软件问题 .....	96	5.5.3 检查与评审 .....	153
4.3.1 标识客户和用户 .....	96	5.5.4 改进 .....	153
4.3.2 理解业务背景 .....	98	5.6 评审分析模型 .....	153
4.3.3 策划并实施需求调查 .....	99	5.7 需求规约 .....	154
4.3.4 定义软件系统的轮廓 .....	100	5.8 需求验证 .....	156
4.4 创建框架用例 .....	102	小结 .....	157
4.4.1 策划并实施用例调查 .....	102	习题 .....	157
4.4.2 以框架用例记录调查结果 .....	103	<b>第6章 软件设计概论</b> .....	159
4.4.3 创建用例图 .....	105	6.1 软件设计的概念 .....	159
4.4.4 整合并评审框架用例 .....	106	6.1.1 软件设计模型 .....	159
4.5 精化用例 .....	106	6.1.2 设计模型的质量要素 .....	161
4.5.1 用例交互动作序列的描述方法 .....	107	6.2 软件设计的基本原则 .....	161
4.5.2 分解或合并用例 .....	111	6.2.1 抽象与逐步求精 .....	162
4.5.3 构建完整用例 .....	112	6.2.2 模块化 .....	165
4.5.4 精化用例图 .....	113	6.2.3 信息隐藏 .....	168
4.5.5 精化业务规则及非功能需求 .....	114	6.2.4 关注点分离 .....	169
4.6 评审用例模型 .....	115	6.3 软件设计的过程模型 .....	171
小结 .....	115	6.3.1 软件设计活动 .....	171
习题 .....	116	6.3.2 迭代式设计过程模型 .....	172
<b>第5章 需求分析与验证</b> .....	117	6.3.3 设计过程模型的裁剪 .....	173
5.1 分析模型的表示 .....	117	小结 .....	173
5.1.1 顺序图 .....	118	习题 .....	174
5.1.2 通信图 .....	123	<b>第7章 软件体系结构设计</b> .....	175
5.1.3 状态图 .....	125	7.1 软件体系结构的概念 .....	176
5.1.4 扩充机制 .....	130	7.1.1 何谓体系结构 .....	176
5.2 需求分析的过程模型 .....	133	7.1.2 体系结构视图 .....	177
5.3 需求优先级分析 .....	134	7.2 体系结构的表示 .....	177
5.3.1 确定需求项优先级 .....	135	7.2.1 包图 .....	178
5.3.2 编排用例分析的优先顺序 .....	136	7.2.2 构件图 .....	180
5.4 用例分析 .....	137	7.2.3 部署图 .....	183
5.4.1 精化领域概念模型 .....	138	7.2.4 对象图 .....	186

7.3 体系结构设计的过程模型 .....	187	第9章 软件详细设计 .....	238
7.4 体系结构设计模式 .....	188	9.1 详细设计的任务与过程模型 .....	238
7.4.1 何谓设计模式 .....	188	9.2 用例设计 .....	240
7.4.2 通用的体系结构模式 .....	190	9.2.1 设计用例实现方案 .....	241
7.5 概念设计 .....	194	9.2.2 构造设计类图 .....	243
7.5.1 关键需求辨识 .....	194	9.2.3 整合并优化用例实现方案 .....	246
7.5.2 体系结构初创 .....	195	9.3 子系统设计 .....	246
7.6 体系结构精化 .....	198	9.3.1 确立内部设计元素 .....	246
7.6.1 逻辑视图体系结构的精化 .....	198	9.3.2 导出设计类图 .....	248
7.6.2 开发视图体系结构的设计 .....	208	9.3.3 设计状态图与活动图 .....	249
7.6.3 物理视图体系结构的设计 .....	210	9.4 构件设计 .....	250
7.6.4 运行视图体系结构的设计 .....	211	9.4.1 为复用而设计构件 .....	250
7.6.5 数据视图体系结构的设计 .....	212	9.4.2 设计构件的定制机制 .....	250
7.6.6 软件体系结构文档 .....	214	9.4.3 设计构件的组装机制 .....	251
7.7 基于构件的体系结构设计 .....	215	9.5 类设计 .....	252
7.7.1 软件复用概述 .....	215	9.5.1 精化类间关系 .....	252
7.7.2 构件创立 .....	216	9.5.2 精化属性和操作 .....	255
7.7.3 构件复用 .....	218	9.5.3 设计状态图与活动图 .....	258
7.8 体系结构验证 .....	219	9.6 数据模型设计 .....	260
小结 .....	220	9.6.1 确定持久数据条目 .....	260
习题 .....	221	9.6.2 确定持久数据的组织结构 .....	260
第8章 人机交互设计 .....	222	9.6.3 确立持久数据操作 .....	264
8.1 以用户为中心的设计 .....	222	9.6.4 优化持久数据操作的性能 .....	264
8.1.1 以用户为中心的设计理念 .....	222	9.7 设计整合与验证 .....	265
8.1.2 人类信息处理模型 .....	223	9.7.1 设计规约 .....	265
8.1.3 以用户为中心的设计方法 .....	224	9.7.2 设计验证 .....	267
8.2 用户界面设计的基本原则 .....	224	小结 .....	267
8.3 用户界面设计模型的表示 .....	225	习题 .....	269
8.4 用户界面设计的过程模型 .....	226	第10章 软件实现 .....	271
8.5 用户及任务分析 .....	227	10.1 软件实现的任务 .....	271
8.5.1 用户分析 .....	227	10.2 软件实现过程 .....	271
8.5.2 任务分析 .....	228	10.3 软件实现与程序设计语言 .....	273
8.6 用户界面的概念设计 .....	229	10.3.1 程序设计语言的演变和分类 .....	273
8.7 用户界面流设计 .....	232	10.3.2 程序设计语言的基本机制 .....	274
8.8 用户界面的精化 .....	234	10.3.3 程序设计语言的选择 .....	275
小结 .....	236	10.4 编程实现 .....	276
习题 .....	236	10.4.1 将编程作为问题求解 .....	276

10.4.2	程序设计范型 .....	277	12.3.2	黑盒测试 .....	336
10.4.3	编程标准 .....	280	12.4	软件测试活动及实施策略 .....	338
10.4.4	编程风格 .....	281	12.4.1	单元测试 .....	338
10.4.5	极限编程与结对编程 .....	282	12.4.2	集成测试 .....	341
10.5	软件调试 .....	283	12.4.3	确认测试 .....	344
10.5.1	调试过程 .....	283	12.4.4	系统测试 .....	345
10.5.2	软件缺陷的分类 .....	284	12.5	面向对象软件的测试 .....	346
10.5.3	调试方法 .....	286	12.5.1	类的测试 .....	346
10.5.4	调试技术 .....	287	12.5.2	交互测试 .....	347
小结	.....	289	12.5.3	继承的测试 .....	347
习题	.....	289	小结	.....	348
<b>第 11 章 结构化软件开发</b>	.....	291	习题	.....	348
11.1	面向数据流的分析方法 .....	291	<b>第 13 章 软件维护</b>	.....	352
11.1.1	数据流图与数据字典 .....	291	13.1	软件维护与进化的概念 .....	352
11.1.2	实体—关系图 .....	293	13.2	软件维护的过程模型 .....	353
11.1.3	数据流图的实时系统扩充 .....	294	13.2.1	结构化与非结构化的维护 .....	353
11.1.4	基于数据流的分析方法 .....	297	13.2.2	维护的成本 .....	354
11.2	面向数据流的设计方法 .....	304	13.2.3	可能存在的问题 .....	355
11.2.1	基本概念和设计过程 .....	304	13.3	可维护性 .....	355
11.2.2	变换分析 .....	305	13.3.1	影响可维护性的因素 .....	356
11.2.3	事务分析 .....	310	13.3.2	若干量化的测度 .....	356
11.3	实时系统设计 .....	314	13.3.3	保证可维护性的复审 .....	356
11.3.1	实时系统性能要求 .....	314	13.4	维护活动及实施策略 .....	357
11.3.2	实时系统设计要素 .....	315	13.4.1	维护组织 .....	357
11.3.3	实时系统设计方法 .....	316	13.4.2	维护的报告与评估 .....	358
11.3.4	设计实例 .....	317	13.4.3	维护 workflow .....	358
小结	.....	320	13.4.4	保存维护记录 .....	360
习题	.....	321	13.4.5	评价维护活动 .....	360
<b>第 12 章 软件测试</b>	.....	324	13.5	维护的副作用 .....	360
12.1	软件测试的概念 .....	324	13.6	逆向工程与软件重构 .....	362
12.1.1	软件测试的任务 .....	324	13.6.1	文档重构 .....	362
12.1.2	测试阶段的信息流程 .....	325	13.6.2	重组 .....	363
12.1.3	测试用例及其设计 .....	325	13.6.3	逆向工程 .....	363
12.1.4	软件测试的原则 .....	327	13.6.4	再工程 .....	365
12.2	软件测试的过程模型 .....	327	小结	.....	366
12.3	软件测试方法 .....	329	习题	.....	366
12.3.1	白盒测试 .....	329	<b>第 14 章 Web 软件工程</b>	.....	368

14.1 基于 Web 的软件及其特性 .....	368	15.4.2 软件质量度量的三层次模型 .....	406
14.1.1 相关概念 .....	368	15.4.3 Boehm、FURPS 和 ISO 9126 度量 模型 .....	410
14.1.2 Web 软件的特性 .....	370	15.4.4 软件质量的评价准则度量 .....	414
14.2 Web 软件的需求工程 .....	371	15.5 软件可靠性度量 .....	418
14.3 Web 软件体系结构设计 .....	372	15.5.1 软件可靠性的概念 .....	418
14.4 Web 界面设计 .....	374	15.5.2 软件修复和软件有效性 .....	419
14.4.1 为全球范围内不同的用户而 设计 .....	374	15.5.3 软件可靠性估算 .....	420
14.4.2 面向客户端浏览器的界面设计 .....	375	15.6 制定软件度量大纲的方法和工具 .....	423
14.4.3 导航设计 .....	376	15.6.1 制定软件度量大纲的方法 .....	423
14.4.4 美工设计 .....	377	15.6.2 软件度量工具 .....	424
14.5 Web 软件的详细设计 .....	377	小结 .....	425
14.5.1 表现层设计 .....	377	习题 .....	425
14.5.2 Web 层设计 .....	379	<b>第 16 章 软件项目管理与过程改进</b> .....	<b>427</b>
14.5.3 业务层设计 .....	381	16.1 软件项目管理与过程模型 .....	427
小结 .....	382	16.1.1 软件项目的概念 .....	427
习题 .....	383	16.1.2 软件项目的过程模型 .....	428
<b>第 15 章 软件度量与估算</b> .....	<b>384</b>	16.1.3 软件项目的原则 .....	428
15.1 软件测量、度量与估算的概念 .....	384	16.1.4 软件项目制品 .....	430
15.1.1 软件测量的意义和作用 .....	384	16.2 软件项目度量与估算 .....	433
15.1.2 度量、测量和估算 .....	385	16.2.1 软件项目分解 .....	434
15.1.3 软件工程测量、度量与估算的基 本内容 .....	387	16.2.2 采用代码行、功能点度量的工作量 估算 .....	436
15.1.4 软件工程测量估算的基本方法 .....	388	16.2.3 软件项目的生产率度量 .....	438
15.2 软件规模度量 .....	390	16.2.4 一般经验估算模型 .....	438
15.2.1 代码行度量 .....	390	16.2.5 COCOMO 模型 .....	439
15.2.2 功能点度量 .....	392	16.2.6 COCOMO II 模型 .....	444
15.2.3 代码行度量与功能点度量的 比较 .....	394	16.2.7 Putnam 模型 .....	446
15.2.4 对象点度量 .....	395	16.3 风险分析 .....	447
15.2.5 软件复用的度量 .....	397	16.3.1 风险标识 .....	448
15.3 软件复杂性度量 .....	398	16.3.2 风险估算 .....	448
15.3.1 软件复杂性及度量原则 .....	398	16.3.3 风险评价和管理 .....	450
15.3.2 控制结构的复杂性度量 .....	399	16.4 软件项目计划 .....	454
15.3.3 体系结构的复杂性度量 .....	401	16.4.1 任务分配与工程进度 .....	454
15.4 软件质量度量 .....	404	16.4.2 任务分解与并行化 .....	454
15.4.1 软件质量的概念 .....	405	16.4.3 工作量分布 .....	455
		16.4.4 工程进度安排 .....	456

---

16.5 软件项目人员和组织 .....	459	16.7.2 软件配置管理的任务 .....	470
16.5.1 软件项目人员 .....	459	16.7.3 软件配置管理标准 .....	474
16.5.2 软件项目团队 .....	460	16.7.4 配置管理工具 .....	474
16.5.3 软件项目的人员组织 .....	461	16.8 软件过程改进 .....	476
16.5.4 评审或估算专家 .....	462	16.8.1 CMM 的基本概念 .....	477
16.5.5 软件项目的人员分布 .....	462	16.8.2 能力成熟度模型 CMM .....	478
16.6 软件质量保证 .....	463	16.8.3 能力成熟度模型集成 CMMI .....	484
16.6.1 软件开发标准 .....	463	16.8.4 CMM 和 CMMI 的选择和应用 .....	489
16.6.2 软件质量保证活动 .....	465	小结 .....	490
16.6.3 基于统计的软件质量保证 .....	466	习题 .....	490
16.7 软件配置管理 .....	467	参考文献 .....	492
16.7.1 软件配置管理的概念 .....	468		

# 第1章 软件与软件工程

由于微电子、计算机、软件、网络 and 多媒体技术的飞速发展，人类社会正从工业社会向信息社会迅速转变。因特网和数亿台计算机进入了办公室、教室、实验室和千家万户。无处不在的无线通信网络连接人们的手机和手提电脑。在软件的支持下形成一个世界范围的、开放的、内容丰富、功能强大、使用便捷、数亿人参与的分布计算环境，实现了全球范围内便捷的信息和知识交流。信息的形式不仅有数字、文字、软件，而且还有图形、图像、声音和视频。个人计算机和电话、电视、汽车、飞机一样，已成为人们工作、生活不可或缺的工具。今天的微处理器和个人计算机已广泛应用于科学研究、教育、办公、企事业管理、工业控制、家用电器、休闲娱乐、武器装备等人类社会的所有领域。大型、巨型计算机系统正担负着数值天气预报、石油地震数据处理、航天发射、电子政务、电子商务、银行信息处理、军事指挥和控制等任务。计算机软件处理、储存、控制的信息和知识已成为当今社会运行和发展的关键。近二三十年来软件产业迅猛发展，已成为和石油、汽车等传统产业并驾齐驱的支柱产业。软件已成为促进科技进步和经济发展的新的驱动力，帮助人们进入了知识经济和信息社会的新时代。

然而，信息社会也不全是美好的，不法分子利用网络环境和计算机产生的负面影响令人忧虑。计算机病毒、计算机犯罪日益猖獗，计算机安全受到了普遍的关注。为了应对这种局面，人们不得不耗费大量的资源研发安全软件、反病毒软件。这些软件的运行耗费了大量的计算机资源，在很大程度上抵消了计算技术进步带给人类的丰硕成果。目前计算机安全已成为软件产业和计算学科的重要组成部分。

社会的信息化程度越高，对软件的依赖性就越强。随着微电子技术、计算机技术、显示技术、传感器技术的革命性变化，人们对计算机软件的功能、性能、质量、安全、成本和交付时间等提出了越来越高的要求。然而不幸的是，人们的软件开发能力总是显得力不从心，大型软件项目的开发计划往往一拖再拖，成本失去控制、软件质量得不到保证的现象仍然时有发生。20世纪60年代末期以来，人们开始借鉴工程的方法开发软件，不仅重视软件开发方法、工具和环境的研究和应用，而且还十分重视软件、软件开发过程的标准化，重视软件的度量、软件人才培养、软件开发团队建设和软件开发过程的管理。40多年来，软件开发能力伴随着硬件的发展不断增强，但人们需要软件产品的功能和性能又有了新的提高。加速软件产业和软件工程学科的发展，不断满足日益增长的软件需求，已成为全社会普遍关注的热点问题。

这一章介绍软件和软件工程的基本概念，包括软件、软件工程、软件开发过程与模型、软件工具与环境，以及与软件工程密切相关的人的因素和基于计算机的系统等。

## 1.1 软件的概念

软件工程的任务是按照客户需求开发软件制品。本节讨论软件的基本概念,包括:定义、组成、软件生存周期、软件的特点、几类常用软件、软件安全与保护。

### 1.1.1 软件与软件的组成

随着计算机的普及和广泛应用,与“硬件”对应的“软件”这个词,从 20 世纪 60 年代初期开始被广泛使用。常见的解释有:软件是计算机系统的重要组成部分;软件是逻辑产品,需要计算机硬件的支撑;软件是计算机控制系统的指挥中枢;软件是信息转换器,它能对信息进行加工、处理或变换;软件是科学、技术、知识和文化的载体;软件是工具,在社会的经济、军事、政治、文化、科学技术、教育和人们的生活、工作中发挥着巨大的作用,等等。

计算机科学对软件的定义是,“软件是在计算机系统支持下,能够完成特定功能和性能的程序、数据和相关的文档”。于是,软件可形式化地表示为:

$$\text{软件} = \text{知识} + \text{程序} + \text{数据} + \text{文档}$$

程序是用计算机程序设计语言描述的。无论是低级语言,如汇编语言,还是高级语言,如 C++、Java,程序都可以在相应语言编译器的支持下映照成操纵计算机硬件执行的代码,或在解释器的支持下操纵计算机硬件的运行。

数据是程序加工的对象和结果。计算机直接加工的数据结构只有简单的整型数、浮点数、逻辑量、字符,人们可以根据需要,在此基础上定义复杂的数据结构。基于大量数据处理的软件需要数据库系统的支持,涉及数据的加工、储存、检索、传输、应用等。

文档记录软件开发的活动和中间制品,记录软件的配置及变更,用于软件专业人员和用户的交流,用于软件开发、过程管理和运行阶段的维护。

国内外标准化组织为软件制定了多种标准。软件的程序、数据和文档应遵循相应的标准。在软件开发和维护过程中应适应软件的变更,保持程序、数据和文档的一致性。一致性是软件质量的重要内容,是软件理解、开发、维护的基础。大型软件开发和维护的一致性需要软件工具和环境的支持。

值得指出的是,软件虽然由程序、数据和文档组成,但软件蕴含着“完成特定功能和性能”的知识和经验。软件的功能和性能除了取决于软件自身的质量外,还取决于软件描述的领域知识和经验。软件具有技术和文化双重属性。很难想象一个结构良好,不能解决实际问题的软件会有什么社会和经济价值。因此软件开发团队在注重软件开发方法的同时,更要注重求解问题的领域知识和经验。没有相关的领域知识和经验,软件就不能完成预定的功能和性能。因此,软件开发团队必须善于与领域专家合作,妥善处理知识产权问题。同时,应有自己的特色和定位,长

期在某一领域进行软件开发,掌握核心技术,提供优质服务,会在竞争中形成明显优势,如石油地震数据处理软件、中长期天气预报软件、银行业务处理软件、医疗设备控制软件等。

软件的描述依赖于软件开发方法、使用的程序设计语言和计算设备。面向过程的软件开发方法使用 Fortran、C 程序设计语言等,面向对象的软件开发方法使用 C++,Java 等程序设计语言。面向机器的汇编语言依赖于处理机的体系结构和 CPU 的指令系统等。

从软件获取的渠道看,可分为通用软件和定制软件两类。通用软件是软件开发机构开发的,公开出售,可以独立使用,如操作系统、办公软件、绘图软件、媒体播放软件等。定制软件是软件承包商应客户要求专门开发的,如金融机构的业务处理软件、机场空中指挥控制软件等。客户可根据情况决定购买或定制。软件可以采用软件制品、软件服务和作为基于计算机的应用系统的一部分这 3 种方式为客户服务并参与市场竞争。

### 1.1.2 软件生存周期

软件和其他产品一样,也有生存周期。软件生存周期是软件从概念形成、进化、运行到退役的全过程。其中,软件进化包括软件开发和维护。软件生存周期各个阶段的详细划分受软件开发过程的影响。传统的软件生存周期划分为软件需求、设计、编码、测试、运行和维护、退役几个阶段,与软件开发过程的阶段划分一一对应(见图 1.1)。软件生存周期各阶段的制品——需求规约、需求模型、软件体系结构的概要设计、详细设计、源程序、可执行代码等反映了软件从概念、雏形(需求)到最终制品的进化过程。软件生存周期关注软件制品本身及其进化的状态,而不是软件制品如何进化的过程。1.3 节将深入讨论软件的进化过程。



图 1.1 传统软件生存周期

现代多数大型复杂的软件系统不能给出软件的全部需求,需求扩展和变更不可避免,有的需求可以准确的确定,有的暂时不能确定,有的甚至还没想到。软件开发从需求确定的部分先做,暂时不能确定的后做。这样,整个软件的生存周期的各个阶段就不能截然划分了,部分软件已完成需求、设计、编码和测试,而另一些正处于需求、设计阶段,还有一些正在确定需求,等等。传统软件生存周期的概念与软件开发过程的一一对应关系发生了本质变化,软件生存周期的概念和作用在逐渐淡化。

### 1.1.3 软件的特点

软件是抽象的逻辑产品,而不是物理产品。由于不受材料的限制,也不受物理定律或加工过程的制约,具有极大的灵活性。软件的灵活性具有双重性,程序员通过编程可以让计算机巧妙地



工作,同时也很容易让软件变得极为复杂,难以理解。软件开发过程的监督、控制、管理有着特殊的困难。因此,软件在开发、生产、维护和使用等方面与硬件相比存在明显的差异。硬件生产可以从市场上买到几乎所有的元器件、部件,然后到工厂组装。经过严格测试、试验、试用后,设计过程中的错误一般是能够排除的。软件开发与硬件开发相比,更依赖于开发人员的业务素质、智力、人员的组织、合作和管理。软件开发的成本和进度很难估计。目前,多数场合软件开发采用定制和采购商用软件相结合的方式,尽量避免从零开始。软件交付前,尽管经过了严格的测试和试用,但仍不能保证软件没有潜伏的缺陷。硬件试制成功之后,批量生产需要建生产线,投入大量的人力、物力和资金。生产过程中还要加强管理和质量控制,对每件产品进行严格的检验。硬件交付后,初期会暴露产品设计、制造中的问题,各零部件需要磨合,经长期使用会发生磨损、老化,于是故障率曲线呈“浴缸”形(见图 1.2(a))。而软件开发完成后,只需对原版软件进行复制即可实现批量生产。但是,软件在使用过程中的维护工作却比硬件复杂得多。首先,软件潜伏的缺陷在运行期间可能暴露,出现运行错误,于是就要进行“纠错性维护”;其次,用户有时需要提高和完善软件的性能,必须对软件制品进行修改,这就要进行“完善性维护”;最后,由于支撑软件制品运行的硬件或软件环境的改变和更新,也需要对软件制品进行修改,进行“适应性维护”。软件的逻辑关系复杂、理解困难,软件在维护过程中很可能产生新的缺陷(见图 1.2(b))。

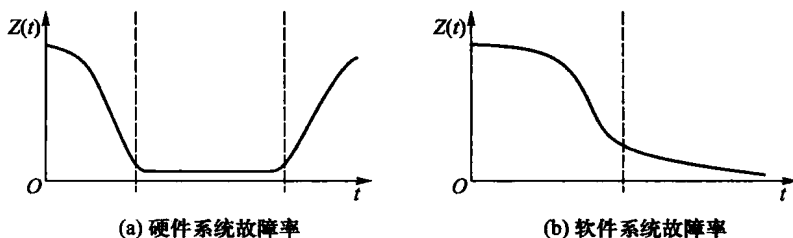


图 1.2 软、硬件系统的故障率曲线

软件的突出优点是不会磨损和老化。只要需要,一个久经考验的优质软件可以长期使用下去,而这一点硬件是做不到的。今天,几乎没有一个用户在使用第一代的电子管计算机,但却有相当多的用户仍在使用 FORTRAN 语言编写的应用程序。很多计算机用户在选择新机型时,提出的一个重要条件往往是原有的应用程序必须能在新机型的支撑环境下运行并有较高的性能/价格比。软件的这一特性构成了一种特殊的文化现象。

#### 1.1.4 几类常用的软件

60 多年来,人们开发了大量的软件,积累了丰富的软件资源并广泛地用于社会生活的各个方面。软件的功能、性能、应用领域、规模、使用方式、工作环境相差甚大。控制洗衣机的软件可能只有几千行代码,但运行在个人计算机上的 Windows XP 的代码达到 350 万行。软件的