



“十二五”普通高等教育本科国家级规划教材

# 软件工程

## (第4版)

齐治昌 谭庆平 宁洪 编著

高等教育出版社

国家精品资源共享课程主讲教材

# 软件工程

## (第4版)

### 本书特色：

- ◆ 本书是一本软件工程的综合性教材，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。
- ◆ 本书强调软件中蕴含的领域知识和经验，软件生存周期的阶段划分与软件开发过程分解分开；用统一建模描述语言UML描述RUP过程中的制品；验证与确认贯穿RUP过程的始终；变更管理和配置管理等若干软件工程相关的重要问题，以实例贯穿始终，强调理论与实践相结合。
- ◆ 提供与教材配套的教学视频与电子教案，以加强软件工程课程实习，方便高校师生教学。



### 数字课程网站

网址：<http://abook.hep.com.cn/1871824>

<http://abook.hep.edu.cn/1871824>

数字课程账号 使用说明详见书内数字课程说明页

ISBN 978-7-04-050960-1



9 787040 509601 >

定价：59.00 元



“十二五”普通高等教育本科国家级规划教材

# 软件工程

## (第4版)

齐治昌 谭庆平 宁洪 编著

高等教育出版社·北京

## 内容简介

本书是在第3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,基于计算机的系统和业务过程建模。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、UML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。本书简化了前版中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、持续集成(CI)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或信息类专业高年级本科生教材,也可作为硕士研究生教材及软件开发人员参考书。

## 图书在版编目(CIP)数据

软件工程 / 齐治昌, 谭庆平, 宁洪编著. --4 版

. --北京: 高等教育出版社, 2019. 3

ISBN 978-7-04-050960-1

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(2018)第 258358 号

## Ruanjian Gongcheng

策划编辑 倪文慧  
插图绘制 于博

责任编辑 倪文慧  
责任校对 张薇

封面设计 于文燕  
责任印制 赵义民

版式设计 张杰

出版发行 高等教育出版社  
社 址 北京市西城区德外大街4号  
邮政编码 100120  
印 刷 北京中科印刷有限公司  
开 本 787mm×1092mm 1/16  
印 张 32  
字 数 700千字  
购书热线 010-58581118  
咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.hepmall.com.cn>  
<http://www.hepmall.com>  
<http://www.hepmall.cn>  
版 次 1997年4月第1版  
2019年3月第4版  
印 次 2019年3月第1次印刷  
定 价 59.00元

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 50960-00

## 数字资源使用说明页

本书配套的数字资源使用方法如下：

1. 计算机访问 <http://abook.hep.com.cn/1871824>, 或手机扫描二维码, 下载并安装 Abook 应用。

2. 注册并登录, 进入“我的课程”。

3. 输入教材封底防伪标签上的数字课程账号(20位密码, 刮开涂层可见), 或通过 Abook 应用扫描封底数字课程账号二维码, 完成课程绑定。

4. 单击“进入课程”按钮, 开始本数字课程的学习。

课程绑定后一年为数字课程使用有效期。受硬件限制, 部分内容无法在手机端显示, 请按提示通过计算机访问学习。

如有使用问题, 请发邮件至 [abook@hep.com.cn](mailto:abook@hep.com.cn)。



扫描二维码

下载 Abook 应用

# 前 言

当前,世界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。经论证,2004年ACM、IEEE-CS联合推出了计算教程软件工程卷CCSE 2004,与其相关的软件工程知识体系(SWEBOK)系统地阐明了软件工程学科的地位和知识体,而软件工程教育知识体系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,CC 2004还将软件工程作为其他4门学科的一个关键的知识领域,即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,短短数年间取得了长足进步。

为适应软件工程教育发展的需要,本教材第3版定位于高校本科生和硕士研究生,是一本软件工程的综合性教材。为加强软件工程的课程实习,我们还编写了配套的软件工程实践教程,供教师和学生选用。关于软件工程学科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。良好的软件实践需要理论与实践相结合,需要技术和工具的支持,还需要开发团队的共同努力。第4版在前3版教材的基础上,借鉴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观念、软件开发的统一过程RUP和描述语言UML、软件需求、设计、实现、验证和确认、维护和进化、持续集成、软件度量、软件项目管理、软件组织的过程改进等。

本书强调下列9个问题。

① 软件中蕴含的领域知识和经验。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、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 workflow、部署 workflow、配置和变更管理 workflow、项目管理 workflow、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 在对软件正确性、可靠性要求极高的应用领域,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,还是介绍了持续集成、软件能力成熟度模型集成等方面的内容。

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。

为便于学生学习,系统地掌握软件开发的全过程,本书选用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。

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。

借再版机会,我们对书中部分词汇、疏漏、错误进行了修订,并且引入持续集成的相关内容。希望这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。

针对学习重点和难点,书中使用二维码提供了若干微视频,读者通过手机扫描即可观看。在本课程的国家级精品资源共享课平台上有全程教学视频,建议读者在阅读本书的过程中系统地学习视频课程,以取得更好的学习效果。

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

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

最后,我们衷心感谢复旦大学的钱乐秋教授,他在百忙中认真地审阅了全部书稿,指出了原

稿中许多疏漏和错误,并提供软件质量标准的宝贵资料。我们也再次向为本书再版付出辛勤劳动的高等教育出版社的领导和编辑、所有使用本书进行教学的教师和学生,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意见。

作 者  
2019年2月

# 目 录

<b>第 1 章 软件与软件工程</b> .....	1	1.5.3 软件工程从业人员的职业 道德 .....	34
1.1 软件的概念 .....	2	1.6 软件工具及软件开发环境 .....	35
1.1.1 软件与软件的组成 .....	2	1.6.1 软件工具 .....	35
1.1.2 软件生存周期 .....	3	1.6.2 软件开发环境 .....	38
1.1.3 软件的特点 .....	3	1.7 基于计算机的系统 .....	40
1.1.4 几类常用的软件 .....	4	1.7.1 基于计算机的系统的概念 .....	40
1.1.5 软件的质量 .....	7	1.7.2 基于计算机的系统的功能 组件 .....	42
1.1.6 软件的安全与保护 .....	9	1.7.3 典型的基于计算机的系统 .....	44
1.2 软件工程的定义 .....	10	1.7.4 基于计算机的系统的研制 .....	46
1.2.1 软件工程的定义 .....	10	小结 .....	51
1.2.2 软件工程的发展 .....	12	习题 1 .....	51
1.2.3 软件工程的目标和原则 .....	15	<b>第 2 章 UML 与 RUP 统一过程</b> .....	53
1.2.4 软件开发的主要方法 .....	16	2.1 案例说明 .....	53
1.3 软件过程模型 .....	18	2.2 面向对象的软件开发方法 .....	54
1.3.1 瀑布模型 .....	18	2.2.1 面向对象的概念 .....	54
1.3.2 增量过程模型 .....	21	2.2.2 面向对象方法的优势 .....	55
1.3.3 原型建造模型 .....	22	2.3 UML 概述 .....	56
1.3.4 螺旋模型 .....	23	2.4 RUP 统一过程 .....	58
1.3.5 基于构件的过程模型 .....	25	2.4.1 RUP 软件过程框架 .....	58
1.3.6 通用软件过程模型 .....	25	2.4.2 RUP 的 5 个阶段 .....	60
1.4 敏捷软件开发原则和应用 .....	29	2.4.3 RUP 的 9 个工作流 .....	62
1.4.1 敏捷软件开发原则 .....	29	小结 .....	66
1.4.2 敏捷软件过程的极限编程 实践 .....	29	习题 2 .....	67
1.4.3 敏捷软件开发过程的应用 .....	31	<b>第 3 章 需求工程概论</b> .....	68
1.5 软件工程人的因素 .....	31	3.1 软件需求的概念 .....	69
1.5.1 软件工程教育 .....	32	3.1.1 软件需求的分类 .....	69
1.5.2 软件工程学科 .....	32		

3.1.2 软件需求的质量要素 .....	69	4.5.3 构建完整用例 .....	113
3.2 需求工程的预备知识 .....	70	4.5.4 精化用例图 .....	115
3.2.1 与用户/客户交流的技巧 .....	70	4.5.5 精化业务规则及非功能 需求 .....	116
3.2.2 需求调查的基本方法 .....	71	4.6 评审用例模型 .....	116
3.2.3 需求建模的基本方法 .....	73	小结 .....	117
3.3 需求工程的过程模型 .....	74	习题 4 .....	117
3.3.1 需求工程中的活动 .....	74	<b>第 5 章 需求分析与验证</b> .....	119
3.3.2 迭代式过程模型 .....	76	5.1 分析模型的表示 .....	119
3.3.3 过程模型的裁剪 .....	77	5.1.1 顺序图 .....	120
小结 .....	80	5.1.2 通信图 .....	125
习题 3 .....	80	5.1.3 状态图 .....	127
<b>第 4 章 需求获取</b> .....	84	5.1.4 扩充机制 .....	132
4.1 软件需求的初始表示 .....	84	5.2 需求分析的过程模型 .....	135
4.1.1 用例 .....	84	5.3 需求优先级分析 .....	136
4.1.2 用例图 .....	85	5.3.1 确定需求项优先级 .....	136
4.1.3 用例的表示 .....	88	5.3.2 编排用例分析的优先顺序 .....	138
4.1.4 类图 .....	89	5.4 用例分析 .....	138
4.1.5 活动图 .....	94	5.4.1 精化领域概念模型 .....	139
4.2 需求获取的过程模型 .....	96	5.4.2 设置分析类 .....	140
4.3 定义软件问题 .....	97	5.4.3 构思分析类之间的协作 关系 .....	142
4.3.1 标识客户和用户 .....	98	5.4.4 导出分析类图 .....	147
4.3.2 理解业务背景 .....	99	5.5 利用快速原型辅助需求分析 .....	153
4.3.3 策划并实施需求调查 .....	101	5.6 评审分析模型 .....	154
4.3.4 定义软件系统的轮廓 .....	102	5.7 需求规约 .....	155
4.4 创建框架用例 .....	103	5.8 需求验证 .....	157
4.4.1 策划并实施工用例调查 .....	104	小结 .....	158
4.4.2 以框架用例记录调查结果 .....	105	习题 5 .....	158
4.4.3 创建用例图 .....	107	<b>第 6 章 软件设计概论</b> .....	160
4.4.4 整合并评审框架用例 .....	107	6.1 软件设计的概念 .....	160
4.5 精化用例 .....	108	6.1.1 软件设计模型 .....	161
4.5.1 用例交互动作序列的描述 方法 .....	109	6.1.2 设计模型的质量要素 .....	162
4.5.2 分解或合并用例 .....	112		

6.2 软件设计的基本原则 .....	163	7.6.6 软件体系结构文档 .....	214
6.2.1 抽象与逐步求精 .....	163	7.7 基于构件的体系结构设计 .....	215
6.2.2 模块化 .....	166	7.7.1 软件复用概述 .....	215
6.2.3 信息隐藏 .....	170	7.7.2 构件创立 .....	216
6.2.4 关注点分离 .....	170	7.7.3 构件复用 .....	219
6.3 软件设计的过程模型 .....	172	7.8 体系结构验证 .....	219
6.3.1 软件设计中的活动 .....	172	小结 .....	220
6.3.2 迭代式设计过程模型 .....	173	习题 7 .....	221
6.3.3 设计过程模型的裁剪 .....	174	<b>第 8 章 人机交互设计</b> .....	223
小结 .....	174	8.1 以用户为中心的设计 .....	223
习题 6 .....	175	8.1.1 以用户为中心的设计理念 .....	223
<b>第 7 章 软件体系结构设计</b> .....	176	8.1.2 人类信息处理模型 .....	224
7.1 软件体系结构的概念 .....	177	8.1.3 以用户为中心的设计方法 .....	225
7.1.1 何谓体系结构 .....	177	8.2 用户界面设计的基本原则 .....	225
7.1.2 体系结构视图 .....	178	8.3 用户界面设计模型的表示 .....	226
7.2 体系结构的表示 .....	178	8.4 用户界面设计的过程模型 .....	227
7.2.1 包图 .....	179	8.5 用户及任务分析 .....	228
7.2.2 构件图 .....	181	8.5.1 用户分析 .....	228
7.2.3 部署图 .....	184	8.5.2 任务分析 .....	229
7.2.4 对象图 .....	186	8.6 用户界面的概念设计 .....	230
7.3 体系结构设计的过程模型 .....	187	8.7 用户界面流设计 .....	233
7.4 体系结构设计模式 .....	189	8.8 用户界面的精化 .....	235
7.4.1 何谓设计模式 .....	189	小结 .....	237
7.4.2 通用的体系结构模式 .....	190	习题 8 .....	237
7.5 概念设计 .....	194	<b>第 9 章 软件详细设计</b> .....	239
7.5.1 关键需求辨识 .....	194	9.1 详细设计的任务与过程模型 .....	239
7.5.2 体系结构初创 .....	195	9.2 用例设计 .....	241
7.6 体系结构精化 .....	198	9.2.1 设计用例实现方案 .....	242
7.6.1 逻辑视图体系结构的精化 .....	198	9.2.2 构造设计类图 .....	244
7.6.2 开发视图体系结构的设计 .....	209	9.2.3 整合并优化用例实现方案 .....	247
7.6.3 物理视图体系结构的设计 .....	211	9.3 子系统设计 .....	247
7.6.4 运行视图体系结构的设计 .....	212	9.3.1 确立内部设计元素 .....	247
7.6.5 数据视图体系结构的设计 .....	214	9.3.2 导出设计类图 .....	249

9.3.3 设计状态图与活动图 .....	250	10.5.3 调试方法 .....	287
9.4 构件设计 .....	251	10.5.4 调试技术 .....	288
9.5 类设计 .....	253	小结 .....	289
9.5.1 精化类间关系 .....	253	习题 10 .....	290
9.5.2 精化属性和操作 .....	257	<b>第 11 章 结构化软件开发</b> .....	291
9.5.3 设计状态图与活动图 .....	259	11.1 面向数据流的分析方法 .....	291
9.6 数据模型设计 .....	260	11.1.1 数据流图与数据字典 .....	291
9.6.1 确定持久数据条目 .....	261	11.1.2 实体-关系图 .....	293
9.6.2 确定持久数据的组织结构 .....	261	11.1.3 数据流图的实时系统扩充 .....	294
9.6.3 确立持久数据操作 .....	265	11.1.4 基于数据流的分析方法 .....	299
9.6.4 优化持久数据操作的性能 .....	265	11.2 面向数据流的设计方法 .....	303
9.7 设计整合与验证 .....	266	11.2.1 基本概念和设计过程 .....	303
9.7.1 设计规约 .....	266	11.2.2 变换分析 .....	304
9.7.2 设计验证 .....	268	11.2.3 事务分析 .....	310
小结 .....	268	11.3 实时系统设计 .....	313
习题 9 .....	269	11.3.1 实时系统性能要求 .....	313
<b>第 10 章 软件实现</b> .....	272	11.3.2 实时系统设计要素 .....	314
10.1 软件实现的任务 .....	272	11.3.3 实时系统设计方法 .....	315
10.2 软件实现过程 .....	272	11.3.4 设计实例 .....	316
10.3 软件实现与程序设计语言 .....	274	小结 .....	319
10.3.1 程序设计语言的演变和 分类 .....	274	习题 11 .....	320
10.3.2 程序设计语言的基本机制 .....	275	<b>第 12 章 软件测试</b> .....	323
10.3.3 程序设计语言的选择 .....	276	12.1 软件测试的概念 .....	323
10.4 编程实现 .....	277	12.1.1 软件测试的任务 .....	323
10.4.1 将编程作为问题求解 .....	277	12.1.2 测试阶段的信息流程 .....	324
10.4.2 程序设计范型 .....	278	12.1.3 测试用例及其设计 .....	324
10.4.3 编程标准 .....	280	12.1.4 软件测试的原则 .....	326
10.4.4 编程风格 .....	281	12.2 软件测试的过程模型 .....	326
10.4.5 极限编程与结对编程 .....	283	12.3 软件测试方法 .....	328
10.5 软件调试 .....	284	12.3.1 白盒测试 .....	328
10.5.1 调试过程 .....	284	12.3.2 黑盒测试 .....	335
10.5.2 软件缺陷的分类 .....	285	12.4 软件测试活动及实施策略 .....	337
		12.4.1 单元测试 .....	337

12.4.2 集成测试 .....	339	14.4.2 集成测试自动化 .....	374
12.4.3 确认测试 .....	343	14.4.3 确认测试自动化 .....	375
12.4.4 系统测试 .....	343	14.5 代码质量分析 .....	376
12.5 面向对象软件的测试 .....	345	14.6 发布与部署 .....	377
12.5.1 类的测试 .....	345	小结 .....	379
12.5.2 交互测试 .....	345	习题 14 .....	379
12.5.3 继承的测试 .....	346	<b>第 15 章 软件度量与估算</b> .....	380
小结 .....	346	15.1 软件测量、度量与估算的 概念 .....	380
习题 12 .....	347	15.1.1 软件测量的意义和作用 .....	380
<b>第 13 章 软件维护</b> .....	350	15.1.2 度量、测量和估算 .....	381
13.1 软件维护与进化的概念 .....	350	15.1.3 软件工程测量、度量与 估算的基本内容 .....	383
13.2 软件维护的过程模型 .....	351	15.1.4 软件工程测量估算的基本 方法 .....	384
13.2.1 结构化与非结构化的维护 .....	351	15.2 软件规模度量 .....	386
13.2.2 维护的成本 .....	352	15.2.1 代码行度量 .....	386
13.2.3 可能存在的问题 .....	353	15.2.2 功能点度量 .....	388
13.3 可维护性 .....	353	15.2.3 代码行度量与功能点 度量的比较 .....	390
13.4 维护活动及实施策略 .....	355	15.2.4 对象点度量 .....	391
13.5 维护的副作用 .....	358	15.2.5 软件复用的度量 .....	393
13.6 逆向工程与软件重构 .....	359	15.3 软件复杂性度量 .....	394
小结 .....	363	15.3.1 软件复杂性及度量原则 .....	394
习题 13 .....	364	15.3.2 控制结构的复杂性度量 .....	395
<b>第 14 章 持续集成</b> .....	365	15.3.3 体系结构的复杂性度量 .....	397
14.1 持续集成概述 .....	365	15.4 软件质量度量 .....	400
14.1.1 持续集成的相关概念 .....	365	15.4.1 软件质量的概念 .....	401
14.1.2 持续集成的过程与管道 .....	367	15.4.2 软件质量度量的三层次 模型 .....	402
14.1.3 持续集成的价值 .....	368	15.4.3 Boehm、FURPS 和 ISO 9126 度量模型 .....	406
14.2 版本控制 .....	369	15.4.4 软件质量的评价准则度量 .....	410
14.2.1 集中与分布式版本控制 .....	369		
14.2.2 版本控制的主线与分支 .....	369		
14.2.3 功能开关 .....	370		
14.3 构建 .....	371		
14.4 测试 .....	373		
14.4.1 单元测试自动化 .....	373		

15.5 软件可靠性度量 .....	414	16.4.1 任务分配与工程进度 .....	450
15.5.1 软件可靠性的概念 .....	414	16.4.2 任务分解与并行化 .....	450
15.5.2 软件修复和软件有效性 .....	415	16.4.3 工作量分布 .....	451
15.5.3 软件可靠性估算 .....	416	16.4.4 工程进度安排 .....	452
15.6 制定软件度量大纲的方法和工具 .....	419	16.5 软件项目人员和组织 .....	455
15.6.1 制定软件度量大纲的方法 .....	419	16.5.1 软件项目人员 .....	455
15.6.2 软件度量工具 .....	420	16.5.2 软件项目团队 .....	456
小结 .....	421	16.5.3 软件项目的人员组织 .....	457
习题 15 .....	421	16.5.4 评审或估算专家 .....	458
<b>第 16 章 软件项目管理与过程改进</b> .....	<b>423</b>	16.5.5 软件项目的人员分布 .....	458
16.1 软件项目管理与过程模型 .....	423	16.6 软件质量保证 .....	459
16.1.1 软件项目管理的概念 .....	423	16.6.1 软件开发标准 .....	459
16.1.2 软件项目管理的过程模型 .....	424	16.6.2 软件质量保证活动 .....	461
16.1.3 软件项目管理的原则 .....	424	16.6.3 基于统计的软件质量保证 .....	462
16.1.4 软件项目制品 .....	426	16.7 软件配置管理 .....	464
16.2 软件项目度量与估算 .....	429	16.7.1 软件配置管理的概念 .....	464
16.2.1 软件项目分解 .....	430	16.7.2 软件配置管理的任务 .....	466
16.2.2 采用代码行、功能点度量的 工作量估算 .....	432	16.7.3 软件配置管理标准 .....	470
16.2.3 软件项目的生产率度量 .....	434	16.7.4 配置管理工具 .....	470
16.2.4 一般经验估算模型 .....	434	16.8 软件过程改进 .....	472
16.2.5 COCOMO 模型 .....	435	16.8.1 CMM 的基本概念 .....	473
16.2.6 COCOMO II 模型 .....	440	16.8.2 能力成熟度模型 CMM .....	474
16.2.7 Putnam 模型 .....	442	16.8.3 能力成熟度模型集成 CMMI .....	480
16.3 风险分析 .....	444	16.8.4 CMM 和 CMMI 的选择和 应用 .....	484
16.3.1 风险标识 .....	444	小结 .....	486
16.3.2 风险估算 .....	445	习题 16 .....	486
16.3.3 风险评价和管理 .....	446	参考文献 .....	488
16.4 软件项目计划 .....	450		

# 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，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，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。

本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,包括软件与软件的组成、软件生存周期、软件的特点、几类常用软件、软件的质量、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类型只有简单的整型数、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类型。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,以及软件开发、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、开发及维护的基础。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但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和领域专家合作,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 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。

从获取的渠道看,软件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、编码与测试、运行与维护、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。

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,有的需求可以准确地确定,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



图 1.1 传统软件生存周期