第一图书网, tushu007.com

<<软件工程>>

图书基本信息

书名：<<软件工程>>

13位ISBN编号：9787040324884

10位ISBN编号：7040324881

出版时间：2012-5

出版时间：齐治昌、谭庆平、 宁洪 高等教育出版社 (2012-05出版)

作者：齐治昌 等 著

页数：499

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

《面向21世纪课程教材：软件工程（第3版）》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。

作者简介

齐治昌教授，博士生导师。
1965年毕业于中国科学技术大学应用数学系。
主要教学和科研领域为软件工程。
40多年来在国防科学技术大学计算机学院参加并完成银河Ⅰ巨型计算机、441BⅢ型计算机FORTRAN语言编译软件、随机振动控制系统软件等多项软件工程项目。
主持和参与多项国家863、国家自然科学基金、国防预研等项目获国家教学优秀成果一、二等奖各1项，部委级科技进步一、二等奖6项。
参与国防科学技术大学计算机学院国家精品课程“软件工程”，计算机科学技术专业国家教学团队的建设工作

<<软件工程>>

书籍目录

第1章 软件与软件工程 1.1 软件的概念 1.1.1 软件与软件的组成 1.1.2 软件生存周期 1.1.3 软件的特点 1.1.4 几类常用的软件 1.1.5 软件的质量 1.1.6 软件的安全与保护 1.2 软件工程的概念 1.2.1 软件工程的定义 1.2.2 软件工程的发展 1.2.3 软件工程的目标和原则 1.2.4 软件开发的主要方法 1.3 软件过程模型 1.3.1 瀑布模型 1.3.2 增量过程模型 1.3.3 原型建造模型 1.3.4 螺旋模型 1.3.5 基于构件的过程模型 1.3.6 通用软件过程模型 1.4 敏捷软件开发原则和应用 1.4.1 敏捷软件开发原则 1.4.2 敏捷软件过程的极限编程实践 1.4.3 敏捷软件开发过程的应用 1.5 软件工程人的因素 1.5.1 软件工程教育 1.5.2 软件工程学科 1.5.3 软件工程从业人员的职业道德 1.6 软件工具及软件开发环境 1.6.1 软件工具 1.6.2 软件开发环境 1.7 基于计算机的系统 1.7.1 基于计算机的系统的概念 1.7.2 基于计算机的系统的功能组件 1.7.3 典型的基于计算机的系统 1.7.4 基于计算机的系统的研制 小结 习题 第2章 UML与RUP统一过程 2.1 案例说明 2.2 面向对象的软件开发方法 2.2.1 面向对象的概念 2.2.2 面向对象方法的优势 2.3 UML概述 2.4 RUP统一过程 2.4.1 RUP软件过程框架 2.4.2 RUP的5个阶段 2.4.3 RUP的9个工作流 小结 习题 第3章 需求工程概论 3.1 软件需求的概念 3.1.1 软件需求的分类 3.1.2 软件需求的质量要素 3.2 需求工程的预备知识 3.2.1 与用户交流的技巧 3.2.2 需求调查的基本方法 3.2.3 需求建模的基本方法 3.3 需求工程的过程模型 3.3.1 需求工程中的活动 3.3.2 迭代式的过程模型 3.3.3 过程模型的裁剪 小结 习题 第4章 需求获取 4.1 软件需求的初始表示 4.1.1 用例 4.1.2 用例图 4.1.3 用例的表示 4.1.4 类图 4.1.5 活动图 4.2 需求获取的过程模型 4.3 定义软件问题 4.3.1 标识客户和用户 4.3.2 理解业务背景 4.3.3 策划并实施需求调查 4.3.4 定义软件系统的轮廓 4.4 创建框架用例 4.4.1 策划并实施用例调查 4.4.2 以框架用例记录调查结果 4.4.3 创建用例图 4.4.4 整合并评审框架用例 4.5 精化用例 4.5.1 用例交互动作序列的描述方法 4.5.2 分解或合并用例 4.5.3 构建完整用例 4.5.4 精化用例图 4.5.5 精化业务规则及非功能需求 4.6 评审用例模型 小结 习题 第5章 需求分析与验证 5.1 分析模型的表示 5.1.1 顺序图 5.1.2 通信图 5.1.3 状态图 5.1.4 扩充机制 5.2 需求分析的过程模型 5.3 需求优先级分析 5.3.1 确定需求项优先级 5.3.2 编排用例分析的优先顺序 5.4 用例分析 5.4.1 精化领域概念模型 5.4.2 设置分析类 5.4.3 构思分析类之间的协作关系 5.4.4 导出分析类图 5.5 利用快速原型辅助需求分析 5.5.1 分析与规划 5.5.2 设计与实现 5.5.3 检查与评审 5.5.4 改进 5.6 评审分析模型 5.7 需求规约 5.8 需求验证 小结 习题 第6章 软件设计概论 6.1 软件设计的概念 6.1.1 软件设计模型 6.1.2 设计模型的质量要素 6.2 软件设计的基本原则 6.2.1 抽象与逐步求精 6.2.2 模块化 6.2.3 信息隐藏 6.2.4 关注点分离 6.3 软件设计的过程模型 6.3.1 软件设计活动 6.3.2 迭代式设计过程模型 6.3.3 设计过程模型的裁剪 小结 习题 第7章 软件体系结构设计 7.1 软件体系结构的概念 7.1.1 何谓体系结构 7.1.2 体系结构视图 7.2 体系结构的表示 7.2.1 包图 7.2.2 构件图 7.2.3 部署图 7.2.4 对象图 7.3 体系结构设计的过程模型 7.4 体系结构设计模式 7.4.1 何谓设计模式 7.4.2 通用的体系结构模式 7.5 概念设计 7.5.1 关键需求辨识 7.5.2 体系结构初创 7.6 体系结构精化 7.6.1 逻辑视图体系结构的精化 7.6.2 开发视图体系结构的设计 7.6.3 物理视图体系结构的设计 7.6.4 运行视图体系结构的设计 7.6.5 数据视图体系结构的设计 7.6.6 软件体系结构文档 7.7 基于构件的体系结构设计 7.7.1 软件复用概述 7.7.2 构件创立 7.7.3 构件复用 7.8 体系结构验证 小结 习题 第8章 人机交互设计 8.1 以用户为中心的设计 8.1.1 以用户为中心的设计理念 8.1.2 人类信息处理模型 8.1.3 以用户为中心的设计方法 8.2 用户界面设计的基本原则 8.3 用户界面设计模型的表示 8.4 用户界面设计的过程模型 8.5 用户及任务分析 8.5.1 用户分析 8.5.2 任务分析 8.6 用户界面的概念设计 8.7 用户界面流设计 8.8 用户界面的精化 小结 习题 第9章 软件详细设计 9.1 详细设计的任务与过程模型 9.2 用例设计 9.2.1 设计用例实现方案 9.2.2 构造设计类图 9.2.3 整合并优化用例实现方案 9.3 子系统设计 9.3.1 确立内部设计元素 9.3.2 导出设计类图 9.3.3 设计状态图与活动图 9.4 构件设计 9.4.1 为复用而设计构件 9.4.2 设计构件的定制机制 9.4.3 设计构件的组装机制 9.5 类设计 9.5.1 精化类间关系 9.5.2 精化属性和操作 9.5.3 设计状态图与活动图 9.6 数据模型设计 9.6.1 确定持久数据条目 9.6.2 确定持久数据的组织结构 9.6.3 确立持久数据操作 9.6.4 优化持久数据操作的性能 9.7 设计整合与验证 9.7.1 设计规约 9.7.2 设计验证 小结 习题 第10章 软件实现 10.1 软件实现的任务 10.2 软件实现过程 10.3 软件实现与程序设计语言 10.3.1 程序设计语言的演变和分类 10.3.2 程序设计语言的基本机制 10.3.3 程序设计语言的选择 10.4 编程实现 10.4.1 将编程作为问题求解 10.4.2 程序设计范型 10.4.3 编程标准 10.4.4 编程风格 10.4.5 极限编程与结对编程 10.5 软件调试 10.5.1 调试过程 10.5.2 软件缺陷的分类 10.5.3 调试方法 10.5.4 调试技术 小结 习题 第11章 结构化软件开发 11.1 面向数据流的分析方法 11.1.1 数

据流图与数据字典 11.1.2 实体一关系图 11.1.3 数据流图的实时系统扩充 11.1.4 基于数据流的分析方法 11.2 面向数据流的设计方法 11.2.1 基本概念和设计过程 11.2.2 变换分析 11.2.3 事务分析 11.3 实时系统设计 11.3.1 实时系统性能要求 11.3.2 实时系统设计要素 11.3.3 实时系统设计方法 11.3.4 设计实例 小结 习题 第12章 软件测试 12.1 软件测试的概念 12.1.1 软件测试的任务 12.1.2 测试阶段的信息流程 12.1.3 测试用例及其设计 12.1.4 软件测试的原则 12.2 软件测试的过程模型 12.3 软件测试方法 12.3.1 白盒测试 12.3.2 黑盒测试 12.4 软件测试活动及实施策略 12.4.1 单元测试 12.4.2 集成测试 12.4.3 确认测试 12.4.4 系统测试 12.5 面向对象软件的测试 12.5.1 类的测试 12.5.2 交互测试 12.5.3 继承的测试 小结 习题 第13章 软件维护 13.1 软件维护与进化的概念 13.2 软件维护的过程模型 13.2.1 结构化与非结构化的维护 13.2.2 维护的成本 13.2.3 可能存在的问题 13.3 可维护性 13.3.1 影响可维护性的因素 13.3.2 若干量化的测度 13.3.3 保证可维护性的复审 13.4 维护活动及实施策略 13.4.1 维护组织 13.4.2 维护的报告与评估 13.4.3 维护工作流 13.4.4 保存维护记录 13.4.5 评价维护活动 13.5 维护的副作用 13.6 逆向工程与软件重构 13.6.1 文档重构 13.6.2 重组 13.6.3 逆向工程 13.6.4 再工程 小结 习题 第14章 Web软件工程 14.1 基于Web的软件及其特性 14.1.1 相关概念 14.1.2 Web软件的特性 14.2 Web软件的需求工程 14.3 Web软件体系结构设计 14.4 Web界面设计 14.4.1 为全球范围内不同的用户而设计 14.4.2 面向客户端浏览器的界面设计 14.4.3 导航设计 14.4.4 美工设计 14.5 Web软件的详细设计 14.5.1 表现层设计 14.5.2 Web层设计 14.5.3 业务层设计 小结 习题 …… 第15章 软件度量与估算 第16章 软件项目管理与过程改进 参考文献

章节摘录

版权页： 插图： 2.4.3 RUP的9个工作流 RUP过程分为软件支持过程和软件生产过程两类。
软件生产过程由业务建模、需求、设计、实现、验证和确认（V&V）、部署工作流组成；软件支持过程由配置和变更管理、项目管理、环境工作流组成。
两类过程都有明确的任务，专业人员通过工作流（即一系列专门活动）生成软件的中间和最终制品。
1.业务建模工作流 业务建模工作流对软件用户的业务流程调研并建模。
不同软件项目业务建模的难度、方式和工作量差异很大。
有的系统需求已经过充分论证，如基于计算机的应用系统，需要软件实现的功能、输入和输出接口比较明确。
系统工程专家或系统工程师可以提供正式的文档或模型。
软件开发人员要学习有关的领域知识，理解软件需求，在此基础上，与系统工程师协商，做一些必要的修改和优化即可完成业务建模。
另一种极端的情况是，用户对开发的系统大致清楚，或部分清楚，部分若明若暗。
这时软件开发人员要与客户一起导出需求，要走访、调研，找到类似系统进行分析对比，必要时还要开发原型。
借助一系列业务流程、"用例"、原型的帮助获取需求、统一认识，进而确定需求。
这样的软件需求获取实际上是一个和用户共同学习的过程，工作量大、时间长、需求变更不可避免。
需求获取和建模是这类软件开发的关键和难点。
2.需求工作流 RUP采用迭代和增量开发方式。
定义软件需求时要识别基本需求，收集、确定、描述用例是定义软件需求的重要组成部分。
软件需求工作流包括需求分析与描述、需求验证等活动。
需求规约完成后，应组织需求验证。
需求验证多采用评审方式，开发机构准备好需求文档，聘请专家、客户、系统分析人员、项目负责人、开发人员代表参加。
对需求文档是否符合用户要求要做出明确结论，对提交给客户的需求文档和提交给开发人员的需求规约的一致性做技术认证，并给出明确意见。
3.设计工作流 软件设计工作流包括软件体系结构设计（相当于传统的软件概要设计）、软构件选取和设计两部分。
软件体系结构也称软件架构，是根据软件需求规约建立的。
大型软件系统往往采用自顶向下或自底向上的方式进行体系结构设计，给出软件系统的总体结构。
软构件选取和设计通常指定义软构件的功能、接口和界面，给出选取或设计软构件的决策。
软件架构具有层次性、模块化的特点，容易分解成易于处理的子系统或构件。
这里的软构件选取和设计通常指定义软构件的功能、接口和界面，给出选取或设计软构件的决策。
在面向对象的方法中可采用"包"或"类"的机制描述、封装子系统或构件，有利于存储、访问控制和配置管理。
子系统或构件之间的交互可用"包"或"类"之间的依赖关系描述。
软件设计要给出每个类的操作和属性描述，提供架构设计说明书、数据库说明书、组装测试计划等文件。
与软件需求工作流、设计工作流相比，软件实现工作流将有更多的人员参与，软件开发工作量明显增大，需求、设计的缺陷蔓延到后续过程将会给编码、测试和集成带来麻烦，造成废品和返工，因此软件设计完成后必须对设计文档进行严格验证和确认，验证和确认设计与需求的一致性、正确性、文档的规范性等，发现问题及时纠正。
4.实现工作流 软件实现工作流包括构件细化、编码、测试、构件集成和集成测试。
构件细化形成若干可编程的程序模块。
根据设计风格选用适宜的程序设计语言描述程序模块的内部细节。
例如，结构化设计可采用C语言，面向对象的设计可采用C++或Java语言。

程序模块的内部细节包括算法、数据结构和各程序模块之间的详细接口。
编写源代码过程要提供必要的说明，建立"模块开发卷宗"，拟定模块测试方案，并对程序模块进行测试。
软件实现活动应遵循软件工程的目标和原则，必须与软件需求和架构保持一致。
5.验证和确认（V&V）工作流 软件验证和确认（V&V）工作流包括对软件原型和文档的审查，对软构件或子系统的测试，交付软件版本的确认测试。
为了保证软件开发的正确性和一致性，避免软件缺陷和错误的蔓延，软件验证和确认（评审和测试）贯穿软件开发的全过程，以便及时反馈、及时修改。
必须对软件开发过程的关键制品进行审查，如需求规约、软件设计文档等。
必须对可运行的软件模块和代码及时进行测试，如单元测试和集成测试，必须对发布版本进行确认测试。
确认测试将按照承诺的系统运行场景（用例）制定软件确认测试计划，测试系统是否实现了用例描述的场景，是否满足系统需求。
确认测试应向用户提交最终的用户手册、操作手册、源程序清单及其他软件文档。
确认测试结束时应生成确认测试报告和项目开发总结报告。
确认测试要采用多种方法和工具、组织专门的测试队伍并严格组织实施。
只有经过严格测试的软件才能保证开发软件的质量。
由专家、客户、软件开发人员组成的软件评审小组通过软件测试结果和确认报告后，软件制品正式得到确认，新的版本可以对外发布，并交付用户使用。
对于一般软件，用户的使用过程也是软件的继续测试过程。
特别是最初发布的β版软件系统社会已有共识，即通过软件用户的试用帮助开发团队发现软件缺陷，开发团队非常希望得到这些信息。
对于要求极高的软件就没有这样的机会了，必须强化质量保证体系、强化测试过程，不允许软件出现灾难性的错误。

编辑推荐

《面向21世纪课程教材:软件工程(第3版)》的特点是在前两版的基础上，借鉴软件工程知识体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。
着重强调以下问题：软件中蕴含的领域知识和经验；将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RIP过程的始终；变更管理和配置管理等，以实例贯穿始终，强调理论与实践相结合。
提供与教材配套的电子教案；与《面向21世纪课程教材:软件工程(第3版)》配套的《软件工程实践教程》已由高等教育出版社正式出版，以加强软件工程课程实习，方便高校师生教学。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com