

<<现代软件工程>>

图书基本信息

书名：<<现代软件工程>>

13位ISBN编号：9787302261391

10位ISBN编号：7302261393

出版时间：2011-10

出版时间：清华大学出版社

作者：张晓龙 编

页数：263

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>

## <<现代软件工程>>

### 内容概要

本书系统地讲述了现代软件工程的基本概念、原理和现代软件方法学。

本书由12章组成。

首先介绍了软件工程相关概念，面向对象方法学的基本概念、面向对象的开发过程以及面向对象分析和面向对象设计技术；接着介绍了uml建模技术以及各种用于可视化建模的模型与图例。

第5~8章讲述了基于软件复用的高级软件工程，包括软件复用的基本概念，基于组件及组件复用，软件设计模式，软件体系结构及其在软件工程中的应用。

第9章介绍了敏捷软件过程，强调敏捷软件过程在软件工程中的作用。

第10章介绍软件测试技术与工具，最后两章分别介绍了软件项目管理，以及基于cmm / cmmi / tsp / psp的软件过程。

本书适合计算机及计算机相近专业的本科生和研究生作为学习软件工程的教材，也适合软件开发人员及其他有关人员作为自学的参考书或培训教材。

## &lt;&lt;现代软件工程&gt;&gt;

## 书籍目录

## 第1章 现代软件工程概述

- 1.1 软件
- 1.2 软件危机
- 1.3 软件工程
- 1.4 软件的生命周期
- 1.5 软件工程的目标和原则
- 1.6 本章小结
- 习题1

## 第2章 面向对象软件开发方法

- 2.1 面向对象基本思想
- 2.2 面向对象基本概念
- 2.3 面向对象方法
- 2.4 面向对象开发统一过程
- 2.5 本章小结
- 习题2

## 第3章 面向对象分析与设计技术

- 3.1 面向对象分析技术
- 3.2 面向对象设计技术
- 3.3 ooa与ood的关系
- 3.4 本章小结
- 习题3

## 第4章 uml建模技术

- 4.1 面向对象建模及uml简介
- 4.2 用例视图
- 4.3 动态模型图
- 4.4 静态模型图
- 4.5 本章小结
- 习题4

## 第5章 软件复用基础

- 5.1 软件复用概述
- 5.2 支持软件复用的软件工程
- 5.3 本章小结
- 习题5

## 第6章 基于组件与web service的软件开发技术

- 6.1 corba组件系统
- 6.2 com+组件系统
- 6.3 j2ee组件系统
- 6.4 web service基础
- 6.5 本章小结
- 习题6

## 第7章 软件设计模式

- 7.1 软件设计模式基础
- 7.2 设计模式的分类
- 7.3 创建型(creational)设计模式
- 7.4 结构型(structural)设计模式

## &lt;&lt;现代软件工程&gt;&gt;

7.5 行为型(behavioral)设计模式

7.6 drawcli中设计模式的应用

7.7 本章小结

习题7

## 第8章 软件体系结构

8.1 软件体系结构基础

8.2 基本的软件体系结构风格

8.3 基于软件体系结构的开发模式

8.4 软件体系结构应用实例

8.5 本章小结

习题8

## 第9章 敏捷软件过程

9.1 敏捷实践

9.2 敏捷开发方法

9.3 xp——极限编程

9.4 scrum

9.5 dsdm——动态系统开发方法

9.6 crystal方法

9.7 fdd特性驱动开发

9.8 asd自适应软件开发

9.9 本章小结

习题9

## 第10章 软件测试技术与工具

10.1 常用的软件测试方法

10.2 常见的软件测试类型

10.3 软件测试技术

10.4 自动化测试

10.5 手工测试与自动化测试的应用比较

10.6 常用的软件测试工具

10.7 本章小结

习题10

## 第11章 软件项目管理

11.1 软件项目管理定义及特点

11.2 软件项目计划

11.3 软件项目风险管理

11.4 软件质量管理

11.5 软件配置管理(software configuration management , scm)

11.6 人员的组织与管理

11.7 本章小结

习题11

## 第12章 软件成熟度模型与软件过程

12.1 能力成熟度模型

12.2 能力成熟度模型集成

12.3 个人软件过程

12.4 团队软件过程

12.5 能力成熟度模型与软件过程之间的关系

12.6 本章小结

习题12

## 章节摘录

版权页：插图：软件工程方法学包含三个要素：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；工具是为运用方法而提供的自动的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的框架，它规定了完成各项任务的工作步骤。

大的软件公司和研究机构一直在研究软件工程方法学，而且也提出了很多实际的软件开发方法。

下面简单介绍几种使用得最广泛的软件工程方法学。

1.结构化方法学结构化方法学也称为传统方法学，它采用结构化技术（结构化分析、结构化设计和结构化实现）来完成软件开发的各项任务，并使用适当的软件工具或软件工程环境来支持结构化技术的运用。

这种方法学把软件生命周期的全过程依次划分为若干个阶段，然后顺序地完成每个阶段的任务。

采用这种方法学开发软件的时候，从对问题的抽象逻辑分析开始，一个阶段一个阶段地进行开发。

前一个阶段任务的完成是开始进行后一个阶段工作的前提和基础，而后一阶段任务的完成通常是使前一阶段提出的解法更进一步具体化，加进了更多的实现细节。

每一个阶段的开始和结束都有严格标准，对于任何两个相邻的阶段而言，前一阶段的结束标准就是后一阶段的开始标准。

在每一个阶段结束之前都必须进行正式严格的技术审查和管理复审，从技术和管理两方面对这个阶段的开发成果进行检查，通过之后这个阶段才算结束；如果没通过检查，则必须进行必要的返工，而且返工后还要再经过审查。

审查的一条主要标准就是每个阶段都应该交出“最新式的”（即和所开发的软件完全一致的）高质量的文档资料，从而保证在软件开发工程结束时有一个完整准确的软件配置交付使用。

文档是通信的工具，它们清楚准确地说明了到这个时候为止，关于该项工程已经知道了什么，同时奠定了下一步工作的基础。

此外，文档也起到备忘录的作用，如果文档不完整，那么一定是某些工作忘记做了，在进入生命周期的下一个阶段之前，必须补足这些遗漏的细节。

结构化方法学中的程序设计采用的是结构化程序设计（Structure Programming，SP），是20世纪80年代主要的程序设计方法，其核心是模块化。

SP方法主张使用顺序、选择、循环三种基本结构来嵌套连接成具有复杂层次的“结构化程序”。

SP的要点是“自顶而下，逐步求精”的设计思想，“独立功能，单出、人口”的模块仅用三种（顺序、分支、循环）基本控制结构的编码原则。

自顶向下的出发点是从问题的总体目标开始，抽象底层的细节，先专心构造高层的结构，然后再一层一层地分解和细化。

这种方法使复杂的设计过程变得简单明了，过程的结果也容易做到正确可靠。

目前，结构化方法学仍然是人们在开发软件时使用得十分广泛的软件工程方法学。

这种方法学历史悠久，为广大软件工程师所熟悉，而且在开发某些类型的软件时也比较有效，因此，在相当长一段时期内这种方法学还会有生命力。

## <<现代软件工程>>

### 编辑推荐

《现代软件工程》：教学目标明确，注重理论与实践的结合，教学方法灵活，培养学生自主学习的能力，教学内容先进，加强对毕业设计的指导作用，教学模式完善，提供配套的教学资源解决方案。

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>