

<<软件工程实用教程>>

图书基本信息

书名：<<软件工程实用教程>>

13位ISBN编号：9787302270263

10位ISBN编号：7302270260

出版时间：2012-5

出版时间：清华大学出版社

作者：陈明

页数：373

字数：608000

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>

## <<软件工程实用教程>>

### 内容概要

本书是软件工程方面的教材，主要内容包括软件工程概述、软件项目可行性研究、软件需求工程、软件项目概要设计、软件项目详细设计、面向对象设计、用户界面设计、软件项目编码、软件质量与质量保证、软件项目测试、软件项目交付与维护、软件项目管理、软件建模与UML等。

本教材的特点是内容系统、语言精练、概念准确。

《21世纪计算机科学与技术实践型教程：软件工程实用教程》可作为高等学校软件工程课程的教材，也可作为从事软件工程人员的参考书。

书籍目录

第1章 软件工程概述

1.1 软件

1.1.1 软件的发展

1.1.2 软件的定义

1.1.3 软件的特点

1.1.4 软件分类

1.2 软件工程的內容与方法

1.2.1 软件危机与软件工程的定义

1.2.2 软件工程的基本內容与目标

1.2.3 软件的基本开发方法

1.2.4 软件工程的基本原则

1.3 软件生存周期与软件开发模型

1.3.1 软件生存周期

1.3.2 软件开发模型

1.4 软件工具与环境

1.4.1 软件工具概述

1.4.2 软件开发工具的功能

1.4.3 软件开发工具的特性

1.4.4 软件开发工具的分类

1.4.5 软件开发环境

1.5 软件工程学的知识体系

小结

习题

第2章 软件项目可行性研究

2.1 软件项目可行性研究的任务

2.2 软件项目可行性研究的步骤

2.3 软件项目可行性研究的要素 2.3.1 经济可行性

2.3.2 技术可行性

2.3.3 社会环境可行性

2.4 系统流程图

2.5 成本效益分析

2.5.1 成本估计

2.5.2 费用估计

2.5.3 度量效益的方法

小结

习题

第3章 软件需求工程

3.1 软件需求工程概述

3.1.1 软件需求分类

3.1.2 需求规格说明

3.1.3 需求工程概念

3.1.4 需求工程过程

3.2 需求获取方法

3.3 需求分析的任务与原则

3.3.1 需求分析的任务

## <<软件工程实用教程>>

### 3.3.2 需求分析的原则

### 3.4 需求建模方法

#### 3.4.1 结构化的需求建模方法

#### 3.4.2 数据流图

#### 3.4.3 数据字典

### 3.5 图形工具

#### 3.5.1 层次方框图

#### 3.5.2 Warnier图

#### 3.5.3 IPO图

### 3.6 需求验证

#### 3.6.1 目的与任务

#### 3.6.2 内容与方法

#### 3.6.3 需求评审

### 3.7 需求管理

#### 3.7.1 需求管理的目标

#### 3.7.2 需求管理的原则

#### 3.7.3 需求开发的管理

#### 3.7.4 需求管理活动

### 小结

### 习题

## 第4章 软件项目概要设计

### 4.1 软件体系结构

#### 4.1.1 概述

#### 4.1.2 系统构成

#### 4.1.3 控制模型

#### 4.1.4 模块化分解

#### 4.1.5 领域相关的体系结构

### 4.2 概要设计的任务与过程

#### 4.2.1 概要设计的任务

#### 4.2.2 概要设计的过程

### 4.3 软件设计的概念与原则

#### 4.3.1 模块化与模块独立性

#### 4.3.2 抽象

#### 4.3.3 结构设计原则

#### 4.3.4 软件复用

#### 4.3.5 设计模式

### 4.4 面向数据流的设计方法

#### 4.4.1 基本概念

#### 4.4.2 系统结构图的组成

#### 4.4.3 变换分析

#### 4.4.4 事务分析

#### 4.4.5 设计优化

### 4.5 面向数据结构的设计方法

#### 4.5.1 Jackson系统开发方法

#### 4.5.2 Warnier方法

### 4.6 概要设计文档评审

### 小结

## <<软件工程实用教程>>

习题

### 第5章 软件项目详细设计

#### 5.1 详细设计的任务与原则

##### 5.1.1 详细设计的任务

##### 5.1.2 详细设计的原则

#### 5.2 详细设计的工具

##### 5.2.1 程序流程图

##### 5.2.2 N-S图

##### 5.2.3 PAD图

##### 5.2.4 PDL

##### 5.2.5 HIPO图

##### 5.2.6 详细设计工具的选择

#### 5.3 详细设计规格说明与复审

##### 5.3.1 详细设计说明书

##### 5.3.2 设计复审

小结

习题

### 第6章 面向对象分析与设计

#### 6.1 面向对象方法

##### 6.1.1 面向对象方法概述

##### 6.1.2 面向对象的软件工程

##### 6.1.3 面向对象的基本概念和特征

#### 6.2 面向对象分析

##### 6.2.1 面向对象分析过程与原则

##### 6.2.2 确定对象与类

##### 6.2.3 确定属性

##### 6.2.4 定义服务

##### 6.2.5 对象间通信

#### 6.3 面向对象设计

##### 6.3.1 面向对象设计的概念

##### 6.3.2 面向对象设计的方法

小结

习题

### 第7章 用户界面设计

#### 7.1 用户界面的主要特征

#### 7.2 设计原则

#### 7.3 用户交互

#### 7.4 信息表示

#### 7.5 帮助系统

#### 7.6 界面设计

#### 7.7 界面设计评价

##### 7.7.1 界面设计评价指标

##### 7.7.2 界面设计评价方法

小结

习题

### 第8章 软件项目编码

#### 8.1 程序设计语言

## <<软件工程实用教程>>

- 8.1.1 程序设计语言分类
- 8.1.2 程序设计语言的特点
- 8.1.3 程序设计语言的选择
- 8.2 编码风格
  - 8.2.1 源程序文档化
  - 8.2.2 数据说明
  - 8.2.3 语句结构
  - 8.2.4 输入输出
- 8.3 程序效率
  - 8.3.1 程序效率准则
  - 8.3.2 算法对效率的影响
  - 8.3.3 影响存储器效率的因素
  - 8.3.4 影响输入输出的因素
- 8.4 编程安全
  - 8.4.1 冗余程序设计
  - 8.4.2 防错程序设计
- 8.5 面向对象程序设计步骤
- 8.6 编码优化
- 小结
- 习题
- 第9章 软件质量与质量保证
  - 9.1 软件质量的定义
  - 9.2 影响软件质量的因素
  - 9.3 软件质量保证
    - 9.3.1 软件质量保证的概念
    - 9.3.2 软件质量保证的策略
    - 9.3.3 SQA小组的任务
  - 9.4 软件质量保证活动
  - 9.5 软件评审
    - 9.5.1 设计质量的评审内容
    - 9.5.2 程序质量的评审内容
  - 9.6 软件质量保证的标准
  - 9.7 软件质量评价
    - 9.7.1 软件质量评价体系
    - 9.7.2 软件质量评价标准
  - 9.8 软件质量框架
    - 9.8.1 高质量软件的特性
    - 9.8.2 软件质量框架的组成
  - 9.9 软件开发质量的定量描述
- 小结
- 习题
- 第10章 软件项目测试
  - 10.1 概述
    - 10.1.1 软件测试技术的发展
    - 10.1.2 软件错误与缺陷
    - 10.1.3 软件测试的定义
    - 10.1.4 软件测试的对象

## <<软件工程实用教程>>

- 10.1.5 软件测试的目的
- 10.1.6 软件测试的原则
- 10.1.7 软件测试的复杂性
- 10.1.8 软件开发各阶段的测试
- 10.2 软件测试方法
  - 10.2.1 静态分析
  - 10.2.2 动态测试
  - 10.2.3 人工测试与机器测试
  - 10.2.4 黑盒测试
  - 10.2.5 白盒测试
  - 10.2.6 白盒测试与黑盒测试的比较
- 10.3 单元测试与集成测试
  - 10.3.1 单元测试
  - 10.3.2 集成测试
  - 10.3.3 确认测试
  - 10.3.4 系统测试
  - 10.3.5 终止测试
- 10.4 面向对象测试
  - 10.4.1 面向对象测试基础
  - 10.4.2 面向对象测试模型
  - 10.4.3 类测试
  - 10.4.4 面向对象的集成测试
  - 10.4.5 面向对象的系统测试
  - 10.4.6 面向对象测试与传统测试的比较
- 10.5 测试的设计与实现
  - 10.5.1 测试计划
  - 10.5.2 测试设计
  - 10.5.3 测试执行
  - 10.5.4 测试总结
- 小结
- 习题
- 第11章 软件项目交付与维护
  - 11.1 软件发布与部署
    - 11.1.1 软件产品发布
    - 11.1.2 软件产品实施
  - 11.2 软件演化的特征
    - 11.2.1 软件演化的动态特征
    - 11.2.2 软件体系结构的进化
  - 11.3 软件维护
    - 11.3.1 软件维护的分类与特点
    - 11.3.2 软件维护的步骤
    - 11.3.3 软件的可维护性
    - 11.3.4 软件维护的副作用
  - 11.4 逆向工程和再生工程
- 小结
- 习题
- 第12章 软件项目管理

## <<软件工程实用教程>>

### 12.1 软件项目的特点及软件管理功能

#### 12.1.1 软件项目的特点

#### 12.1.2 软件管理的功能

#### 12.1.3 软件项目的工作范围

### 12.2 软硬件资源

#### 12.2.1 人力资源

#### 12.2.2 硬件

#### 12.2.3 软件

### 12.3 人员的计划和组织

### 12.4 成本估计及控制

#### 12.4.1 软件开发成本估算方法

#### 12.4.2 专家估算法

#### 12.4.3 成本估算模型

### 12.5 进度计划

#### 12.5.1 软件工作的特殊性

#### 12.5.2 各阶段工作量的分配

#### 12.5.3 制定开发进度

### 12.6 软件配置管理

#### 12.6.1 基线

#### 12.6.2 软件配置项

#### 12.6.3 软件配置管理过程

### 12.7 软件管理方案

### 12.8 能力成熟度模型

#### 12.8.1 软件过程评估的必要性

#### 12.8.2 CMM的主要用途

#### 12.8.3 CMM的体系结构

#### 12.8.4 CMM的等级

#### 12.8.5 CMM的内部结构

#### 12.8.6 CMM的应用

### 小结

### 习题

## 第13章 软件建模与UML

### 13.1 模型概述

### 13.2 常用的软件建模模型

### 13.3 需求建模

### 13.4 UML

#### 13.4.1 UML概述

#### 13.4.2 UML的表示法

#### 13.4.3 UML软件开发过程概述

#### 13.4.4 Rational Rose概述

### 13.5 形式化方法

### 小结

### 习题

### 参考文献

## 章节摘录

版权页：插图：1.确定对系统的综合需求 对系统的综合需求主要有系统功能需求、系统性能需求、运行需求、将来可能提出的需求。

系统分析人员与用户协商，澄清模糊需求，删除无法做到的需求，改正错误需求。

对于系统功能需求，应该划分出系统必须完成的所有功能。

而系统性能需求包括响应时间、精确度指标需求、安全性等。

运行需求集中表现为对系统运行时所处环境的需求，如软硬件运行环境限定需求等。

最后，对于将来可能提出的需求，应该明确地列出那些虽然不属于当前系统开发范畴，但是据分析将来很可能会提出来的需求。

这样做的目的是在设计过程中对系统将来可能的扩充和修改做准备，以便于需要时能比较容易地进行这种扩充和修改，更有利于系统维护。

2.分析系统的数据需求 任何一个软件系统本质上都是信息处理系统，系统必须处理的信息和系统应该产生的信息在很大程度上决定了系统的面貌，对软件设计有深远影响。

分析系统的数据需求是由系统的信息流归纳抽象出数据元素组成、数据的逻辑关系，数据字典格式和数据模型，并以输入/处理/输出的结构方式表示。

因此，必须分析系统的数据需求，这是软件需求分析的一个重要任务。

3.导出系统的逻辑模型 就是在理解当前系统“怎样做”的基础上，抽取其“做什么”的本质。

在物理模型中有许多物理因素，但随着分析工作的深入，有些非本质因素就显得不必要了，因而需要对物理模型进行分析，区分本质和非本质因素，去掉那些非本质因素就可获得反映系统本质的逻辑模型。

综合上述两项分析的结果，明确目标系统要“做什么”，可以导出系统的详细逻辑模型。

具体做法是：首先确定目标系统与当前系统的逻辑差别；然后将变化部分看做是新的处理步骤，对功能图（一般为数据流图）及对象图进行调整；最后由外及里对变化的部分进行分析，推断其结构，获得目标系统的逻辑模型。

通常用数据流图、数据字典和主要的处理算法描述这个逻辑模型。

4.修正系统开发计划 在经过需求分析阶段的工作后，分析员对目标系统有了更深入、更具体的认识，因此可以对系统的成本和进度作出更准确的估计，在此基础上应该对开发计划进行修正。

5.开发原型系统 在计算机硬件和许多其他工程产品的设计过程中经常使用样机。

建造样机通常有两个主要目的：检验关键设计方案的正确性及系统是否真正满足用户的需要。

同样，对于软件系统的开发，使用原型系统的主要目的是使用户通过实践获得关于未来的系统将怎样为他们工作的更直接、更具体的概念，从而可以更准确地提出和确定他们的要求。

采取建立原型系统的策略的主要理由如下：由于人类认识能力的局限，不能预先指定所有要求；在用户和系统分析员之间存在固有的通信鸿沟；用户需要一个现实的系统模型，以便获得实践经验；而且在开发过程中重复和反复是必要的和不可避免的。

用户试用了原型系统以后，能够指出系统的哪些特性是他们喜欢的，哪些是他们感到不能接受的，以及他们还需要哪些新的功能。

根据经过实践检验的用户需求而开发出来的系统，更可能真正满足用户的需要。

特别是当所开发的系统是全新的，用户从来也没有使用类似系统的经验时，更应该认真考虑开发原型系统的必要性和可能性。

<<软件工程实用教程>>

编辑推荐

<<软件工程实用教程>>

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>