

<<软件工程实例教程>>

图书基本信息

书名：<<软件工程实例教程>>

13位ISBN编号：9787302238096

10位ISBN编号：730223809X

出版时间：2010-11

出版时间：清华大学出版社

作者：吴洁明 等编著

页数：371

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>

<<软件工程实例教程>>

前言

本书历经两年多的时间，今天终于完成了。

软件工程方面的教材已经非常多了，本书以培养应用型软件人才为目标，突出4个特点：第一，突出案例教学，从始至终贯穿了生活实例和软件项目的案例，帮助读者理解软件工程的内容、掌握软件工程的方法，灵活应用到实际的软件工程项目之中。

第二，内容新颖实用，软件工程本身强调不断改进和完善过程，因此本书充实了软件模式、可扩展标记语言、代码重构等内容，并介绍了这些内容在软件工程中的应用。

第三，可操作性强，在介绍分析和设计方法时，给出了详细的步骤，使初学者可以参照具体的步骤，尽快掌握并应用书中介绍的方法。

第四，不仅告诉读者应该怎么做才是好的，还告诉了读者什么是不好的，为什么不好，以帮助读者深刻地领会软件工程的原理和规范，促使读者在实际工程中自觉应用软件工程方法，自觉遵守软件工程专业规范。

“软件工程”任重而道远，我们应该从两个方面理解“软件工程”：一方面是把“软件项目”或“软件产品”的开发和维护工作当成一个工程去做，也就是说，作为一个“软件工程”的建设者和管理者，我们应该强调规划、设计、实施和验收过程的规范化和文档化；另一方面把“软件工程”作为一门学科，深入研究这门学科存在的问题，找出解决问题的方法，设计解决问题的过程，发明解决问题的工具。

本书共分为14章，第1章介绍软件工程概述，从软件危机的现象入手，介绍经典案例，并且进行详细的点评，由此引起读者对软件危机的重视，对软件工程的渴望。

接着介绍软件的特点，讲述软件工程的基本原理、研究的主要内容，以及软件工程的发展历史。

为了让读者对软件工程学科有一个完整的认识，第1章还介绍了软件工程学科的知识体系，最后介绍了软件工程师的职业素质和从业要注意的十大问题。

<<软件工程实例教程>>

内容概要

本书以培养应用型软件人才为目标，全面系统地阐述了软件工程的基本概念、原理和典型方法。全书突出4个特点：第一，从始至终贯穿案例教学的思想，提高读者的学习兴趣；第二，内容新颖实用，介绍厂软件模式、xml、代码重构等内容在软件工程中的应用；第三，可操作性强，读者可参照书中给出的模板和案例，构建自己的应用；第四，通过对反面案例的点评，帮助读者深刻地领会软件工程的原理和规范，促使读者在实际工程中自觉应用软件工程的方法，自觉遵守软件工程规范。

本书适合作为高等院校“软件工程”课程的教材或参考书，书中给出了大量的表格和模板，可作为软件公司的培训教材使用，对具有一定实践经验的软件工程人员也有很好的参考价值。

<<软件工程实例教程>>

书籍目录

第1章 软件工程概述	1.1 软件危机	1.2 软件工程	1.3 软件工程学科知识体系	1.4 软件工程师职业道德
练习1	第2章 软件过程	2.1 软件过程的概念	2.2 几个典型的软件过程模型	2.3 迭代与递增
2.4 rup软件开发过程	2.5 敏捷开发	练习2	第3章 软件工程管理	3.1 人员组织与管理
3.2 软件规模与成本估算	3.3 软件开发计划与控制	3.4 软件配置管理	3.5 软件质量管理	练习3
第4章 需求工程	4.1 需求工程的概念	4.2 确定系统目标和范围	4.3 需求获取方法	4.4 需求分析的任务
4.5 需求分析的原则	4.6 定义软件的质量属性	4.7 需求优先级	4.8 需求验证技术	4.9 需求管理
练习4	第5章 结构化需求分析	5.1 结构化分析的主要技术	5.2 结构化分析方法的实现步骤	5.3 编写需求规格说明书
5.4 结构化分析案例	练习5	第6章 结构化软件设计	6.1 软件设计的概念	6.2 软件设计原则和影响设计的因素
6.3 结构化设计方法	6.4 小型图书馆图书信息管理系统软件结构设计	6.5 优化软件结构	6.6 数据设计	6.7 详细设计
6.8 设计复查	练习6	第7章 面向对象基础	7.1 从一个例子看结构化方法的问题	7.2 面向对象的
7.3 面向对象开发的方法简介	7.4 uml语言	7.5 用例图	7.6 活动图	7.7 状态图
7.8 交互图	7.9 类图	7.10 配置图	7.11 组件图	练习7
第8章 面向对象分析	8.1 面向对象分析概述	8.2 建立用例模型	8.3 建立对象模型	8.4 构造动态模型
8.5 评审分析模型	8.6 面向对象的需求分析规格说明书	练习8	第9章 面向对象设计	9.1 面向对象的设计概念
9.2 面向对象的设计原则——类设计原则	9.3 xml在软件设计中的应用	9.4 基于uml的面向对象设计过程	9.5 面向对象设计规格说明书	练习9
第10章 用户界面设计	10.1 界面设计原则	10.2 界面设计	10.3 用户界面评价	10.4 用户界面设计案例分析
练习10	第11章 软件编码	11.1 程序设计语言	11.2 良好的编程实践	11.3 软件编码规范
11.4 代码重构	练习11	第12章 软件测试	12.1 软件测试基本概念	12.2 软件测试过程
12.3 软件测试计划	12.4 测试用例设计和测试执行	12.5 单元测试	12.6 集成测试	12.7 系统测试
12.8 验收测试	12.9 界面测试	12.10 面向对象的测试	12.11 软件测试工具	练习12
第13章 软件交付与维护	13.1 软件维护概念	13.2 维护过程	13.3 提高软件的可维护性	练习13
第14章 如何编写软件开发文档	14.1 软件文档的作用和要求	14.2 软件文档的种类和提供时机	14.3 软件文档的编写步骤	14.4 如何写好软件文档
14.5 文档管理	14.6 软件工程标准	14.7 几个常用软件文档的模板	练习14	参考文献

<<软件工程实例教程>>

章节摘录

插图：可维护性——它描述纠正一个缺陷或进行一个变更的容易程度。

可维护性取决于软件的可理解性、软件的结构和选择的软件开发工具。

为了使软件易于维护，通常需要规范设计和实现，例如：函数调用不能超过两层，以便于执行跟踪；每个模块中源代码与注释的比例为2：1；对库存统计报表格式变化的修改时间不能超过一周等类似的定量描述。

可移植性——度量把软件从一种环境移植到另一种环境中所花费的工作量。

为了实现可移植性，需要研究软件要移植的环境。

可移植性与高效性可能会有冲突，为了软件具有更好的可移植性，系统分析人员会做更多的限制，例如为了提高软件的可移植性，尽量不使用运行环境提供的库函数等限制。

可移植性对软件的成功不是最重要的，因此分析人员要与用户一起协商，平衡性能的取舍。

可重用性——表明一个软件组件可用于其他软件的程度。

可重用软件的开发成本会比较高，因为可重用软件必须标准化、资料齐全、不依赖于特定的应用程序和运行环境。

可测试性——指测试软件时查找缺陷的简易程度，如果软件中包含复杂的算法和处理逻辑，或者使用了复杂的数据结构，或者功能模块间的关系复杂，都会影响可测试性。

可测试性需求描述的例子有：一个模块的最大循环复杂度不能超过20。

循环复杂度是衡量一个模块源代码中逻辑分支数目的参数，一个模块中的逻辑分支过多会影响可测试性。

可理解性——是指人们通过阅读程序源代码和相关文档了解程序功能、结构和运行方式的容易程度。

一个可理解的程序应该具备下面的一些特征。

- 模块结构良好，功能完备。
- 程序算法简明，没有含糊不清的代码，使用有意义的过程名。
- 代码风格和设计风格一致。
- 采用结构化编程，程序只使用顺序、分支和循环三种基本语句结构，并且每个模块都是单入口单出口。
- 尽量使用简单的数据结构，使用有意义的数据名和变量名。
- 程序处理完整，程序不仅实现了基本的功能，而且还要有数据检查、出错处理等辅助功能。

<<软件工程实例教程>>

编辑推荐

《软件工程实例教程》：北京市高等教育精品教材立项项目。

<<软件工程实例教程>>

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>