

<<软件工程>>

图书基本信息

书名：<<软件工程>>

13位ISBN编号：9787512404489

10位ISBN编号：7512404484

出版时间：2011-6

出版时间：北京航空航天大学出版社

作者：宋广军，黎明，杜鹃，王崇 编著

页数：203

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>>

内容概要

面对无穷无尽的计算机应用需求，软件开发已成为软件开发人员面临的主要任务。

“软件工程”已成为计算机教学一门重要的专业课。

由宋广军编著的《软件工程》以软件的生命周期为主线，重点讨论结构化的软件开发方法，包括结构化分析、结构化设计、编码、测试。

通过对基本概念、基本原理、基本技术、基本方法的学习，使读者能很快运用工程的方法与技术开发软件。

近些年来面向对象软件开发方法和技术不断普及，用最后两章的篇幅介绍面向对象的基本概念，面向对象的分析和设计方法。

《软件工程》内容尽量做到通俗易懂，图文并茂，原理、方法与实例相结合。

可作为高等学校计算机专业教材，也可供计算机软件人员和计算机用户参考。

<<软件工程>>

书籍目录

第1章 软件工程概述

- 1.1 软件工程与软件危机
 - 1.1.1 软件的发展阶段
 - 1.1.2 软件危机
 - 1.1.3 软件工程
- 1.2 软件开发模型
 - 1.2.1 软件生命周期
 - 1.2.2 软件开发的瀑布模型
 - 1.2.3 原型化开发模型
 - 1.2.4 螺旋模型
 - 1.2.5 增量模型
 - 1.2.6 面向对象生存期模型
 - 1.2.7 喷泉模型
 - 1.2.8 基于四代技术的模型

习题1

第2章 软件计划

- 2.1 问题定义
- 2.2 可行性研究
 - 2.2.1 可行性研究的任务
 - 2.2.2 可行性研究过程
 - 2.2.3 系统流程图
 - 2.2.4 可行性论证报告
- 2.3 成本效益分析
- 2.4 项目开发计划
- 2.5 系统规格说明及评审

习题2

第3章 软件需求分析

- 3.1 需求分析概述
 - 3.1.1 需求分析的基本原则
 - 3.1.2 需求分析的任务
 - 3.1.3 需求分析的步骤
 - 3.1.4 需求规格说明与验证
- 3.2 数据流图(dfd)
 - 3.2.1 符号
 - 3.2.2 命名
 - 3.2.3 特点和用途
 - 3.2.4 数据流图的画法
- 3.3 数据字典
 - 3.3.1 数据字典的内容
 - 3.3.2 定义数据的方法
 - 3.3.3 数据字典的实现
- 3.4 实体—联系图
- 3.5 结构化分析方法
 - 3.5.1 实现的步骤
 - 3.5.2 画分层dfd图的指导原则

<<软件工程>>

3.5.3 结构化分析方法的局限

3.6 结构化分析示例

习题3

第4章 总体设计

4.1 总体设计的任务和过程

4.2 软件设计的基本原理

4.2.1 问题分解

4.2.2 模块化

4.2.3 抽象与逐步求精

4.2.4 信息隐蔽

4.2.5 模块独立性

4.3 总体设计的工具

4.3.1 层次图

4.3.2 ipo图

4.3.3 hipo图

4.4 结构化设计方法

4.4.1 信息流分类

4.4.2 结构图

4.4.3 变换分析

4.4.4 事务分析

4.4.5 混合型分析

习题4

第5章 详细设计

5.1 详细设计的任务和过程

5.2 结构化程序设计思想

5.2.1 对goto语句使用的不同看法

5.2.2 结构化的控制结构

5.3.3 逐步细化的实现方法

5.3 详细设计的工具

5.3.1 程序流程图

5.3.2 盒图(n-s图)

5.3.3 pad图

5.3.4 伪代码和pdl语言

5.3.5 判定表与判定树

5.4 jackson程序设计方法

5.4.1 jackson图

5.4.2 jackson方法

5.5 程序结构复杂度的定量度量

5.5.1 mccabe方法

5.5.2 halstead方法

5.6 人—机界面设计

5.6.1 用户的使用需求分析

5.6.2 人—机界面的设计原则

5.6.3 人—机界面实现的原则

5.7 软件安全问题

5.7.1 软件安全控制的目的

5.7.2 软件错误的典型表现

<<软件工程>>

5.7.3 软件系统安全控制的基本方法

5.7.4 软件的安全控制设计

5.8 软件设计复审

习题5

第6章 程序编码

6.1 编码的目酌

6.2 程序设计语言

6.2.1 程序设计语言分类

6.2.2 程序设计语言的特征属性

6.2.3 程序设计语言的使用准则

6.3 程序设计风格

6.3.1 使用程序内部的文档

6.3.2 数据说明原则

6.3.3 语句构造规则

6.3.4 输入输出准则

6.4 提高效率的准则

6.5 防止编码错误

习题6

第7章 软件的测试

7.1 基本概念

7.1.1 软件测试目标

7.1.2 软件测试的原则

7.1.3 软件测试的方法

7.1.4 软件测试的过程

7.2 软件测试技术

7.2.1 白盒测试

7.2.2 黑盒测试

7.2.3 实用综合测试策略

7.3 软件测试策略

7.3.1 单元测试

7.3.2 集成测试

7.3.3 验收测试

7.3.4 系统测试

7.3.5 软件测试过程

7.4 调试技术

7.4.1 调试过程

7.4.2 调试技术

7.4.3 调试原则

习题7

第8章 软件实施与维护

8.1 软件维护的种类

8.2 软件维护的特点

8.2.1 软件工程与软件维护的关系

8.2.2 影响维护工作量的因素

8.2.3 软件维护的策略

8.2.4 维护的成本

8.2.5 可能存在的问题

<<软件工程>>

8.3 维护任务的实施

8.3.1 维护组织

8.3.2 维护报告

8.3.3 维护过程

8.3.4 维护记录的保存

8.3.5 对维护的评价

8.4 软件的可维护性

8.4.1 软件可维护性定义

8.4.2 影响软件可维护性的因素

8.4.3 提高软件的可维护性方法

8.5 软件维护的副作用

8.5.1 修改代码的副作用

8.5.2 修改数据的副作用

8.5.3 修改文档的副作用

8.6 逆向工程和再工程

8.6.1 预防性维护

8.6.2 逆向工程的元素

习题8

第9章 软件项目管理

9.1 软件工程管理概述

9.1.1 软件工程管理的重要性

9.1.2 管理的目的与内容

9.2 软件工作量估算

9.2.1 软件开发成本估算方法

9.2.2 算法模型估算

9.3 风险管理

9.3.1 风险分析

9.3.2 风险识别

9.3.3 风险估算

9.3.4 风险评估

9.3.5 风险监控

9.4 进度计划

9.4.1 任务的确定与进度计划

9.4.2 gantt图

9.4.3 工程网络技术

9.4.4 项目的追踪和控制

9.5 软件配置管理

9.5.1 软件配置

9.5.2 软件配置管理任务

9.6 软件质量保证与cmm

9.6.1 软件质量

9.6.2 软件指令保证措施

9.6.3 能力成熟度模型cmm

9.6.4 能力成熟度模式整合(cmmi)

习题9

第10章 面向对象方法学与建模

10.1 面向对象方法学的基本概念

<<软件工程>>

- 10.1.1 传统方法学存在的问题
- 10.1.2 面向对象方法学的发展状况
- 10.1.3 面向对象方法学的要素和优点
- 10.2 统一建模语言
 - 10.2.1 模型的建立
 - 10.2.2 uml概述
 - 10.2.3 uml的特点与应用
- 10.3 面向对象分析
 - 10.3.1 面向对象分析
 - 10.3.2 建立对象模型
 - 10.3.3 建立动态模型
 - 10.3.4 功能模型
- 习题10
- 第11章 面向对象设计与实现
 - 11.1 面向对象设计
 - 11.1.1 面向对象设计准则及启发规则
 - 11.1.2 软件重用
 - 11.1.3 对象设计
 - 11.1.4 系统设计
 - 11.2 面向对象的实现
 - 11.2.1 面向对象程序设计语言
 - 11.2.2 面向对象程序设计方法
 - 11.2.3 面向对象程序设计风格
 - 11.2.4 面向对象的软件测试
- 习题11
- 参考文献

<<软件工程>>

章节摘录

版权页：插图：3.运行时期运行时期是软件生命周期的最后一个时期。

其主要工作是作好软件维护。

维护的目的是使软件在整个生命周期内保证满足用户的需求和延长软件的使用寿命。

软件维护的具体活动包括纠错维护、适应性维护、功能性维护和预防性维护。

所谓纠错性维护就是改正在软件运行过程中暴露出来的系统遗留的各种错误。

这种维护活动在系统交付初期比较频繁，但当系统进入稳定运行期后应该很少发生。

适应性维护是指当系统运行环境发生变化以后，为适应这种改变必须对软件进行的修改。

功能性维护是指在软件过程中为满足用户需求的变化与扩充对软件所做的修改。

预防性维护则是指为改善软件将来的可维护性所做的准备工作。

软件运行稳定以后，维护的主要活动应该是适应性和功能性维护。

虽然没有把维护阶段进一步划分成更小的阶段，但是实际上每一项维护活动都应该经过提出维护要求、分析维护要求、提出维护方案、审批维护方案、确定维护计划、修改软件设计、修改程序、测试程序、复查验收等一系列步骤，因此实质上是经历了一次压缩和简化了的软件定义和开发的全过程。

1.2.2软件开发的瀑布模型瀑布模型也称软件生存周期模型，由W.Royce于1970年首先提出。

根据软件生存周期各个阶段的任务，瀑布模型从可行性研究（或称系统分析），逐步进行阶段性变换，直至通过确认测试，并得到用户确认的软件产品为止。

瀑布模型上一阶段的变换结果是下一阶段变换的输入，相邻两个阶段具有因果关系，紧密相连。

一个阶段工作的失误将蔓延到以后的各个阶段。

为了保障软件开发的正确性，每个阶段任务完成后，都必须对它的阶段性产品进行评审，确认之后再转入下一阶段的工作。

评审过程发现错误和疏漏后，应该反馈到前面的有关阶段修正错误、弥补疏漏，然后再重复前面的工作，直至某一阶段通过评审后再进入下一阶段。

这种形式的瀑布模型是带有反馈的瀑布模型。

严格按照软件生命周期的阶段划分，顺序执行各阶段构成软件开发的瀑布型模型。

它是传统的软件工程生存期模式。

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>