

<<软件工程原理>>

图书基本信息

书名：<<软件工程原理>>

13位ISBN编号：9787040369069

10位ISBN编号：7040369060

出版时间：2013-2

出版时间：沈备军、陈昊鹏、陈雨婷 高等教育出版社 (2013-02出版)

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>

书籍目录

第一部分 软件工程基础 第1章 绪论 1.1 软件 1.1.1 软件的作用和特性 1.1.2 软件的发展和面临的新挑战 1.1.3 软件危机 1.2 工程 1.2.1 软件开发与艺术、科学和工程的关系 1.2.2 工程的定义和特性 1.2.3 工程的准则 1.3 软件工程 1.3.1 软件工程的知识和知识域 1.3.2 软件工程的金三角 1.3.3 控制软件开发的复杂性 1.3.4 软件工程经济学 1.4 案例概述 本章小结 参考文献 习题1 第2章 软件过程 2.1 软件过程的概念 2.1.1 软件过程的定义和作用 2.1.2 软件过程的分类 2.1.3 软件过程的核心元素 2.2 软件生命周期模型 2.2.1 瀑布模型 2.2.2 增量模型 2.2.3 演化模型 2.3 统一软件过程 2.3.1 阶段 2.3.2 迭代 2.3.3 过程规范 2.4 敏捷过程 2.4.1 敏捷过程的价值观和原则 2.4.2 XP 2.4.3 Scrum 2.4.4 MSF 2.5 软件过程的选择、裁剪、评估和改进 2.5.1 软件过程的选择和裁剪 2.5.2 软件过程的评估 2.5.3 软件过程的改进 本章小结 参考文献 习题2 第3章 软件建模 3.1 模型和建模 3.1.1 模型 3.1.2 软件模型的3个层次：CIM、PIM和PSM 3.1.3 软件建模方法 3.2 结构化方法 3.2.1 结构化分析 3.2.2 结构化设计 3.3 面向对象方法 3.3.1 面向对象的基本概念 3.3.2 面向对象的基本原则 3.3.3 面向对象的分析和设计 3.3.4 结构化方法和面向对象方法的比较 3.4 基于构件的开发方法 3.4.1 构件 3.4.2 基于构件的软件开发 3.4.3 构件开发 3.4.4 构件管理 3.4.5 应用开发和构件组装 3.4.6 与面向对象方法的关系 3.5 面向服务方法 3.5.1 面向服务架构的基本概念 3.5.2 面向服务建模的内涵 3.5.3 面向服务建模的方法 3.6 模型驱动开发方法 3.6.1 从代码驱动到模型驱动 3.6.2 元模型和MOF 3.6.3 基于MDA的软件开发方法 3.7 形式化方法 3.7.1 形式化方法的概念 3.7.2 形式化规约 3.7.3 形式化开发 3.7.4 形式化验证 3.8 敏捷建模方法 3.8.1 敏捷建模的原则 3.8.2 敏捷建模的实践 本章小结 参考文献 习题3 第4章 需求工程 4.1 软件需求面临的挑战 4.2 需求工程的概念 4.2.1 软件需求 4.2.2 需求的层次 4.2.3 优秀需求应具有的特性 4.2.4 软件需求工程 4.3 需求获取 4.3.1 需求来源 4.3.2 需求获取技术 4.4 需求分析建模 4.4.1 分析模型 4.4.2 分析建模的准则 4.5 需求定义和验证 4.5.1 软件前景文档 4.5.2 软件需求规约 4.5.3 用户界面原型 4.5.4 软件需求验证 4.6 需求管理 4.6.1 定义需求基线 4.6.2 需求变更控制和版本控制 4.6.3 需求跟踪 本章小结 参考文献 习题4 第5章 面向对象的分析建模 5.1 面向对象的分析模型 5.1.1 用例图 5.1.2 活动图 5.1.3 类图 5.1.4 时序图 5.1.5 通信图 5.1.6 包图 5.2 分析建模的步骤 5.3 用例建模 5.3.1 执行者的识别 5.3.2 用例的识别 5.3.3 用例图的构建 5.3.4 用例规约的撰写 5.3.5 用例模型的优化 5.4 建立概念模型 5.5 用例实现的识别 5.6 分析类的识别 5.6.1 13种分析类 5.6.2 识别边界类 5.6.3 识别控制类 5.6.4 识别实体类 5.7 用例分析 5.7.1 职责分配的通用原则 5.7.2 时序图和通信图的构建 5.7.3 类图的构建 本章小结 参考文献 习题5 第6章 设计工程 6.1 设计工程概述 6.1.1 软件设计过程 6.1.2 软件设计模型 6.2 软件设计原则 6.2.1 抽象 6.2.2 分解和模块化 6.2.3 封装和信息隐藏 6.2.4 高内聚和低耦合 6.3 软件质量属性的设计策略 6.3.1 可用性设计策略 6.3.2 可修改性设计策略 6.3.3 性能设计策略 6.3.4 安全性设计策略 6.3.5 可测试性设计策略 6.3.6 易用性设计策略 6.4 架构风格 6.4.1 通用结构的架构风格 6.4.2 分布式系统的架构风格 6.4.3 交互式系统的架构风格 6.4.4 自适应系统的架构风格 6.4.5 其他架构风格 6.5 设计模式 6.5.1 创建型设计模式 6.5.2 结构型设计模式 6.5.3 行为型设计模式 6.6 人机界面设计 6.6.1 界面设计原则 6.6.2 人机交互方式 6.6.3 界面设计过程 6.6.4 界面设计问题 本章小结 参考文献 习题6 第7章 面向对象的设计建模 7.1 面向对象的设计模型 7.1.1 设计建模过程 7.1.2 构件图 7.1.3 部署图 7.1.4 状态机图 7.2 架构设计 7.2.1 14+1架构视图 7.2.2 逻辑视图设计 7.2.3 进程视图设计 7.2.4 开发视图设计 7.2.5 物理视图设计 7.2.6 场景视图设计 7.2.7 视图选择 7.3 包和子系统设计 7.3.1 包设计原则 7.3.2 子系统设计 7.4 类设计 7.4.1 类设计原则 7.4.2 类设计过程 7.5 持久化设计 7.5.1 实体对象模型与数据库设计 7.5.2 数据库设计 7.5.3 持久化框架 本章小结 参考文献 习题7 第8章 软件测试 8.1 软件测试概念 8.1.1 软件测试定义 8.1.2 软件测试原则 8.1.3 软件可测试性 8.2 软件测试策略 8.2.1 软件测试层次 8.2.2 软件质量属性测试 8.2.3 其他软件测试策略 8.3 软件测试技术 8.3.1 测试技术分类 8.3.2 基于直觉和经验的测试 8.3.3 基于代码的测试 8.3.4 基于规约的测试 8.3.5 基于错误的测试 8.3.6 基于模型的测试 8.3.7 基于应用类型的测试 8.4 软件测试过程 8.4.1 测试计划 8.4.2 测试设计 8.4.3 测试开发 8.4.4 测试执行 8.4.5 测试评估 8.5 自动化测试 8.5.1 自动化测试的定义和优势 8.5.2 自动化测试适用的场合 8.5.3 自动化测试的误区 8.5.4 测试工具 本章小结 参考文献 习题8 第9章 软件演化和软件维护 9.1 软件演化 9.1.1 软件演化的定义 9.1.2 软件演化法则 9.2 软件维护基础 9.2.1 软件维护的重要性 9.2.2 软件维护的概念和分类 9.3 软件维护的关键问题 9.3.1 技术问题 9.3.2 管理问题 9.3.3 维护成本估算 9.4 软件维护过程与维护活动 9.4.1 维护过程 9.4.2 维护活动 9.5 软件维护技术 9.5.1 程序理解 9.5.2 逆向工程 9.5.3 再工程 本章小结 参考文献 习题9

第二部分 软件工程管理 第三部分 高级主题 附录 IEEE软件工程知识体系 (SWEBOK)

章节摘录

版权页：插图：软件过程包括：（1）软件实现过程 软件实现过程（Software Implementation Process）用于在软件中产生一个特定的系统元素（即软件项），以满足需求。

它是实现过程在软件中的特化，包括7个过程：软件实现过程、软件需求分析过程、软件架构设计过程、软件详细设计过程、软件构建过程、软件集成过程和软件合格测试过程。

（2）软件支持过程 软件支持过程（Software Support Process）提供一组特定的活动来执行特定的软件过程，支持软件实现过程，以帮助软件项目取得成功。

它包括8个过程：软件文档管理过程、软件配置管理过程、软件质量保证过程、软件验证过程、软件确认过程、软件评审过程、软件审计过程和软件问题解决过程。

（3）软件复用过程 软件复用过程（Software Reuse Process）支持一个组织跨项目复用软件项，它包括3个过程：领域工程过程、复用资产管理过程和复用程序管理过程。

2.1.3软件过程的核心元素 根据软件过程的定义，软件过程的核心元素包括活动、任务、工件和角色。其中活动可细分成子活动，它是任务的集合，任务将输入工件加工成输出工件。

活动和任务是由人来执行的，软件也是为人而开发的，一个人可以担任多个角色，一个角色可以由多人承担。

简单来说，软件过程的核心元素回答了谁（角色）用什么（输入工件、资源）如何（活动、任务）完成工作（输出工件）的问题。

此外，还可以把目标和度量指标作为软件过程的有机组成部分。

（1）活动：是过程的核心，指过程中包含的所有开发、维护和管理等活动，它们通常是并发的和合作的。

一个活动可以细分为多个子活动，从而形成一个树状的层次结构，不同的活动处于不同的抽象层次。

（2）任务：是活动的细分。

例如创建项目计划可以细分为多个任务，识别和确定任务项，确定任务项优先级，评估任务项工作量，安排任务项完成次序，确定任务项完成人员等。

（3）工件：是软件过程的工作产品，可以分为输入工件和输出工件。

工件涉及软件工程产生的各种工作内容，甚至是软件过程本身。

例如，代码、测试案例、需求等都是工件，是某个活动或任务的输入或输出。

（4）角色：定义了软件工程组织的环境中，个人或协同工作的多人小组的行为和职责。

角色代表项目中个人承担的任务，并定义其如何完成工作。

（5）资源：是指执行过程所需的最佳实践、工具、机器、技术和场地等支持。

其中最佳实践是一种经过实际检验的方法、准则或者过程，遵循最佳实践通常可以减少项目风险、提高开发效率。

之所以将工具也归为软件过程的元素，是因为开发活动越来越复杂，没有工具的支持，开发活动很难有效进行。

（6）目标：每一个过程都有其明确的目标，例如开发一个新的软件产品，或者对一个已有软件进行版本升级。

过程的活动则可以拥有一个细分的目标，活动间相互协同达到过程的目标。

（7）度量指标：指过程目标的具体测量和分析，如进度、成本、质量、返工率等，从而实现过程的定量管理。

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>