

• 1. 软件的发展问题

软件的发展历史 软件的概念和特点 软件的分类 软件危机的现象和原因

• 2. 软件工程与软件过程

软件工程的概念 软件工程的基本目标 软件过程与软件生命周期 常见的软件开发模型及优缺点

• 3. 软件开发的基本策略

软件复用的基本思想 分而治之的策略 优化与折衷的方法



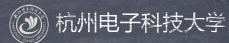
- 1 可行性研究的任务
- 2 可行性研究过程
- 3 系统流程图

- 4数据流图
- 5 数据字典
- 6 成本/效益分析



- 1 需求分析的任务
- 2 与用户沟通获取需求的方法
- 3 分析建模与规格说明
- 4 实体-联系图

- 5 数据规范化
- 6 状态转换图
- 7 其他图形工具
- 8 验证软件需求



• 概述

- 非形式化方法的缺点
- 形式化规格说明方法的优点
- 形式化规格说明方法的缺点
- 有穷状态机
- Petri网
- Z语言



• 总体设计的目的

- 用比较抽象的方法确定系统概要地是如何实现的(How to do generally!)。
- 从初步的数据流图导出(设计出)软件结构。
- 主要分为系统设计(DFD、方案、功能)和软件结构设计(DFD、软件结构)二个阶段。

• 总体设计的过程——9步

• 设想可能方案、选取合理方案、推荐最佳方案、功能分解细化DFD、设计软件结构、设计数据库、制定测试计划、编写文档、 审查和复审。

• 总体设计的原理

• 模块化、抽象vs逐步求精、信息隐蔽/局部化、模块独立性(耦合、内聚)

• 总体设计的启发规则——7条原则

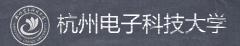
• 保证设计出良好的软件结构。

• 总体设计的图形工具

• 层次图、HIPO图、结构图。

结构化设计方法(SD)

- 区分数据流图的类型(变换型、事务型);
- 结构化设计方法的步骤(7步)



详细设计的关键任务

- 在总体设计的基础上,进一步确定软件怎样具体实现。
- 主要任务包括人机界面设计、模块过程设计。

人机界面设计(指导原则)

- 重视4个人机界面设计问题:系统响应时间、用户帮助设施、出错信息处理、命令交互方式。
- 领会3类人机界面设计指南:一般交互、信息显示、数据输入。

• 过程设计(工具)

• 掌握结构化设计工具(图形、表格、语言):程序流程图、盒图、PAD图、判定表、判定树、PDL、Jackson方法。

程序复杂性度量(方法)

- McCabe方法: 基于程序流中"循环数"的环形复杂度。
- Halstead方法:基于程序中运算符和运算数的程序长度。

第7章 实现



编码

- 编码是将设计结果翻译成程序(某种程序语言编写)。
- 编码风格对软件的可读性、可维护性、可靠性、可用性很重要。
- 编码风格包括: 良好的文档(注释)、数据说明、语句构造、输入输出和效率保障等。

测试

- 测试是主要任务是发现错误并改正错误,保证软件的可靠性。
- 两类基本测试方法: 白盒测试和黑盒测试。
- 测试的过程包括:单元测试(模块)、子系统测试(子系统集成)、系统测试(确认需求)、验收测试(用户参与)、平行运行(试运行)。
- 白盒测试技术:逻辑覆盖测试(语句、判定、条件、判定/条件、条件组合;点、边、路径)、控制结构测试(基本路径、条件、循环)。
- 黑盒测试技术: 等价划分、边值分析、错误推测。

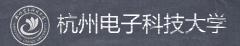
调试

• 寻找错误原因和确定出错位置: 蛮干法、回溯法、原因排除法(对分法、归纳法、演绎法)。

软件可靠性

- 软件可靠性定义
- 软件可用性定义
- 稳态可用性Ass估算
- 平均无故障时间MTTF估算
- 错误数估算方法: 植入错误法、分别测试法(标记错误)

第8章 维护



- 软件的定义和特点
- 软件维护的4类活动 改正性、适应性、完善性、预防性
- 决定软件可维护性的基本要素可理解、可测试、可修改、可移植和可重用性
- 文档是影响软件可维护性的决定因素
- 软件再工程(预防性维护)

选择题 (20分)

判断题 (10分)

简答题 (15分)

综合题 (55分)