软件工程

软件工程概述

- 软件
  - 软件的概念
    - 软件是计算机系统中与硬件相对应的部分，是一组列指令、数据及实现其所定义规则的
    - 通俗的理解：软件是计算机系统中，为了实现某个功能，按照某种顺序编写的一系列指令的集合，上面的程序的操作和使用的文档。
  - 软件的分类
    - 按照范围产品范围的不同
      - 系统软件
      - 支撑软件
      - 应用软件
    - 按照软件产品服务对象的不同
      - 通用软件
      - 定制软件
    - 按照软件产品规模的不同
      - 小型软件
      - 中型软件
      - 大型软件
      - 超大型软件
    - 按照软件产品工作方式的不同
      - 实时软件
      - 分时软件
      - 交互式软件
      - 批处理软件
- 软件危机
  - 软件危机的表现形式
    - 软件开发和维护的困难
    - 软件成本预算的困难与实际需求不符
    - 软件的可靠性差
    - 软件难以维护
    - 软件质量无法满足社会的需求
  - 消除软件危机的途径
    - 使用软件管理的科学方法，提高生产效率
    - 开发出使用软件开发工具，提高生产效率
    - 开发出使用软件的一切，以与现实世界相互联系
    - 在满足现实需求的前提下，提倡软件质量
- 软件工程
  - 软件工程的形成与发展
    - 20 世纪 70 年代
      - 提出了软件生产工程的概念，明确，方法和手段
    - 20 世纪 80 年代
      - 提出了软件生产工程的概念，明确，方法和手段
      - 提出的原则有两种以外人了反用和成果的原
      - 提供开发过程从目标管理和过程管理
    - 21 世纪
      - 智能化与自动化的人的 2001 年聚集一堂，开发第二代最新开发技术
      - 智能软件开发技术.png
      - 通信以计算，大数据，物联网，人工智能，云代表，开放平台等技术
      - "大规模计算"，"智能知识体系的知识"和"软件开发"等的最新可能实现了新不断的工程主题的主要方向
      - 为实现软件的，人工认知生产的"工程化生产"，同时的目，采用工具的概念，原理，技术和方法，使用工具的，程序的过程，程序的过程，使用管理技术的工具和方法，实现高效的工程 (Software Engineering，SE)
  - 软件工程的定义
    - 定义要求的软件问题
    - 软件的可行性研究
    - 开发出高质量的软件
    - 行业和现代的开发成本
    - 用最可信的维护费用，
    - 最新的可信的开发工具，及时支付使用
- 软件生命周期
  - 软件定义
    - 问题定义
    - 可行性分析
    - 需求分析
    - 软件分析
    - 软件开发
    - 概要设计
    - 详细设计
    - 编码和单元测试
    - 综合测试
    - 软件维护
    - 运行维护
    - 软件分析
    - 软件设计
- 软件工程模型
  - 基本模型
  - 快速原型模型
  - 螺旋模型
  - 瀑布模型
  - 统一模型
  - 基于构件的开发模型
  - 敏捷软件开发模型
  - 全球化趋势
  - 开放化趋势
  - 模块化趋势
- 软件工程实施趋势
- 软件工程项目管理
  - 可行性研究的目的：用最小的代价在尽可能短的时间内确定问题是否值得开发，要不要开发
  - 在解决问题之前，必须先搞清楚问题是什么，然后再决定能够采取哪种方案来解决问题
  - 可行性研究的步骤
    - 1. 系统定义
    - 2. 对现行系统进行分析研究
    - 3. 导出新系统的逻辑模型
    - 4. 设计方案
    - 5. 推荐可行的方案
    - 6. 修改可行性方案
    - 7. 制定开发计划
    - 8. 编写可行性研究报告
  - 系统流程图 (System Flowchart)
    - 概念：表明信息在系统各个部分之间的流动情况（不要与入"一"一表示信息的流工程图相对）
    - 作用：写清楚各步骤所发生的对入现在的流程，流程的问题等
    - 基本内容
      - 使用图形符号以直线子形式描述系统内每一个流动（物质，动力，数据等，这些使用，入工过程等）
      - 用"一"表示信息在系统各个部分之间的流动情况（不要与入"一"一表示信息的流工程图相对）
      - 符号
        - 处理
        - 输出/输出
        - 决策
        - 换页连接
        - 数据流
        - 文档
        - 源机/终端
        - 信息
        - 人工输入
        - 人工操作
        - 辅助操作
        - 通信链接
    - 系统流程图的步骤
  - 数据流图 (Data Flow Diagram，DFD)
    - 概念：数据流图是一种图形符号工具，也通过对现实世界的建模来对工具
    - 作用：描述系统的模型部分组成，系统的之间的关系，数据的流向，数据的加工，入和出的输出，的以及数字的信息的流动
    - 基本符号
      - 数据流 (Data Flow)
      - 数据处理 (Process)
      - 数据存储 (Data Store)
      - 外部实体 (External Entity)
    - 设计步骤
      - 1. 列出系统的输入和输出
      - 2. 画系统内部
      - 3. 为每一个数据流命名
      - 4. 为加工命名
    - 数据流图的步骤
  - 数据字典 (Data Dictionary，DD)
    - 概念：数据字典是结构化分析的另一种电子的工具
    - 作用：为数据流图中每个数据流，数据存储，数据处理入出每项数据流或文件的数据存储提出说明
    - 内容
      - 数据流
      - 数据项
      - 数据存储
      - 数据处理
    - 符号
      - =
      - {}
      - {}
      - {}
    - 数据字典的步骤
- 软件问题定义及可行性分析
- 需求分析
  - 需求分析的定义：软件开发流程上是指软件产品的软件定义，分析产品的对人的需求的内容，还是软件开发的基础的关键，模糊，简单，安全的要求的是在软件的开发的是"软件的"的是的，主要概念
  - 需求分析的步骤：确定软件系统必须完成哪些工作才能满足用户对软件的要求，即需要确定软件的是"做什么"，而非"怎么做"
    - 确定软件系统的总体要求
    - 分析系统的数据要求
    - 建立系统的逻辑模型
    - 修正系统原型计划
    - 编写软件需求规格说明书
    - 需求分析评审
    - 6 个方面的具体任务
  - 需求分析的步骤
    - 需求获取
    - 分析建模
    - 需求规定
    - 需求验证
  - 结构化需求分析方法
    - 结构化分析 (Structured Analysis，SA) 方法
    - 需求的获取
      - 1. 调查
      - 2. 根据的应用领域选择技术
      - 3. 快速原型法
  - 需求分析的图形化工具
    - DFD 图
    - 状态图
    - 层次方框图
    - Warnier 图
    - IPO 图
- 总体设计（软件设计与为：总体与详细设计阶段）
  - 总体设计的目标是：
    - 目标：用抽象模型的方法来确定要系统要完成哪些的任务
    - 任务：把分析阶段得到的数据结构和逻辑模型，转换成适合于计算机实现的物理模型，为详细设计阶段提供可供设计的图纸，要求设计
    - 系统设计阶段：确定系统的具体实现方案
    - 结构设计阶段：确定软件的结构
    - 1. 设想供选择的方案
    - 2. 选取合理的方案
    - 3. 推荐最佳方案
    - 4. 功能分等
    - 5. 设计软件结构
    - 6. 数据设计
    - 7. 指定测试计划
    - 8. 编写文档
    - 9. 审查和复查
  - 总体设计的步骤
    - 模块
      - 接口
      - 功能
      - 逻辑
      - 状态
      - 模块化的形式：确定软件的结构，确定设计模块，确定可理解性
    - 抽象
      - 概念：把事物的共性归纳和概括起来，取其共性，它忽略掉一些不重要的细节，抽象是指抽象事物的本质特性而暂时不考虑它的细节
    - 逻辑
      - 概念：逻辑是一种思维方式，一种推理的方式，一种表达方式
      - 两种逻辑
      - 1. 逻辑是一种思维方式，一种推理的方式，一种表达方式
      - 2. 逻辑是一种思维方式，一种推理的方式，一种表达方式
    - 信息隐藏与局部化
    - 模块独立
      - 概念：模块是一个具有明确的功能的数据和程序的集合，模块是程序的基本单位，模块的大小和复杂程度，模块的内部结构，外部接口，模块的数据结构等等，模块的内部结构，外部接口，模块的数据结构等等，模块的内部结构，外部接口，模块的数据结构等等，
      - 模块
        - 名称
        - 数据
        - 特征
        - 功能
        - 过程
        - 接口
        - 内部结构
        - 从上到下顺序的自上而下的顺序，顺序的自上而下的顺序
      - 内聚
        - 偶然内聚
        - 逻辑内聚
        - 时间内聚
        - 过程内聚
        - 通信内聚
        - 顺序内聚
        - 功能内聚
        - 从上到下顺序的自上而下的顺序，顺序的自上而下的顺序
    - 耦合
    - 类型
    - 内聚
    - 类型
    - 模块化的形式：确定软件的结构，确定设计模块，确定可理解性
    - 1. 抽象的结构，即用解决的问题
    - 2. 模块的概念的定义
    - 3. 深度，宽度，扇入，扇出都应当适中
    - 4. 模块的作用域应该在受控域之内
    - 5. 调用模块的结构的最佳顺序
    - 6. 设计扇入以，单出口的模块
    - 7. 模块功能应该可以预测
  - 软件结构设计原则
  - 结构化设计工具
    - 层次图
    - HIPO 图
    - 层次图和 HIPO 的区别为：层次图用来描述软件的层次结构，其中一个矩形代表一个模块，箭头表示模块的调用关系，HIPO 图是层次图的扩展，HIPO 图用来表示模块的功能，HIPO 图用来表示模块的功能
    - 结构图
    - yourdon 提出的结构图是进行软件结构设计的另一个有力工具，其中一个矩形代表一个模块，方框内的模块的功能名称或主要功能，方框之间的箭头或直线连接表示调用关系
    - 面向数据流的软件结构设计方法
    - 概念：面向过程的流程为体现的设计结构的方法，其以流程的数据流图的流程的设计结构的方法，因此流程的数据流程为主了解软件的方法，软件的具体的流程的设计方法，实现的具体要求。
- 详细设计
  - 详细设计的目标和任务
  - 详细设计的原则
  - 用户界面的设计
  - 详细设计的工具
  - 面向数据结构的设计方法
  - 程序的编码与程序质量管理
- 软件编码测试
  - 程序设计风格
  - 软件测试
  - 软件测试的目标
  - 软件测试的过程
  - 软件测试用例定义
  - 黑盒测试用例设计
  - 白盒测试用例设计
  - 软件测试方法应用策略
  - 软件调试
- 软件维护
  - 软件维护概念
  - 软件可维护性
  - 软件重用和工程

传统软件工程

面向对象软件工程

- 面向对象技术与 UML
  - 面向对象分析概述
  - UML 概述
  - UML 图
- 面向对象分析
  - 面向对象分析概述
  - 面向对象分析过程
  - 面向对象分析原则
  - 建立对象模型
  - 建立动态模型
  - 建立功能模型
- 面向对象设计
  - 面向对象设计概述
  - 面向对象设计过程与准则
  - 面向对象设计的原则
  - 系统设计
  - 对象设计
- 面向对象编码与测试
  - 面向对象设计原则
  - 面向对象设计风格
  - 面向对象测试原则
  - 面向对象测试