

# 1、软件危机

## 概念p1

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。

## 典型表现(背至少三点)p2

- 对软件开发成本和进度的估计常常很不准确
- 用户对“已完成的”软件系统不满意的现象经常发生
- 软件产品的质量往往靠不住
- 软件常常是不可维护的
- 软件通常没有适当的文档
- 软件成本在计算机系统总成本中所占比例逐年上升
- 软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势

## 产生软件危机原因p3

- 软件缺乏“可见性”，软件开发过程的进展情况较难衡量
- 规模庞大，且程序复杂性将随着程序规模的增加而指数上升
- 实践过程中或多或少采用了错误的方法和技术

## 消除软件危机途径p4

- 对计算机软件有正确认识
- 充分认识软件开发不是某种个体劳动的神秘技巧，而是一种组织良好、管理严密、各类人员协同配合、共同完成的项目工程
- 总结开发软件成功的技术和方法，尽快消除一些错误概念和做法
- 开发和使用更好的软件工具

## 软件是程序、数据、相关文档的完整集合p4

## 程序

是能够完成预定功能和性能的可执行的指令序列

## 数据

是指程序能够适当地处理信息的数据结构

## 文档

是开发、使用和维护程序所需要的图文资料

## CASE

---

Computer计算机 Aided辅助 Software软件 Environment环境

## 2、软件工程

### 概念p5

---

软件工程是指导计算机软件开发和维护的一门工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好技术方法结合起来，以经济地开发出高质量的软件并有效地维护它。

### 软件工程本质特性p6

---

- 软件工程关注于大型程序的构造
- 软件工程的中心课题是控制复杂性
- 软件经常变化
- 开发软件的效率非常重要
- 和谐地合作是开发软件的关键
- 软件必须有效地支持它的用户
- 在软件工程领域中通常具有一种文化背景的人替具有另一种文化背景的人创造产品

### 软件工程的基本原理p7

---

- 用分阶段的生命周期计划严格管理
- 坚持进行阶段评审
- 实行严格的产品控制
- 采用现代程序设计技术
- 结果应能清除地审查

- 开发小组的人员应该少而精
- 承认不断改进软件工程实践的必要性

## 软件工程方法学p9

---

- 传统方法学
- 面向对象方法学

## 软件生命周期阶段p12

---

- 问题定义
- 可行性研究
- 需求分析
- 总体设计
- 详细设计
- 编码和单元测试
- 综合测试
- 软件维护

## 什么是软件过程p14

---

软件工程是为了获得高质量软件所需要完成的一系列任务的框架，它规定了完成各项任务的工作步骤。

## 瀑布模型p15

---

### 特点

- 阶段间具有顺序性和依赖性
- 推迟实现的观点
- 质量保证的观点

### 优点

- 可强迫开发人员采用规范的方法
- 严格规定了每个阶段必须提交的文档
- 要求每个阶段交出的所有产品都必须经过质量保证小组的仔细验证

## 缺点

- 瀑布模型是由文档驱动的。在可运行的软件产品交付给用户之前，用户只能通过文档来了解产品是什么样的

## 快速原型模型p16

---

### 特点

- 快速建立起来可以在计算机上运行的程序，完成的功能往往是最终产品能完成的功能的一个子集。

### 优点

- 软件产品的开发基本上是线性顺序进行的

### 缺点

- 书上没找到

## 增量模型p17

---

### 特点

- 把软件产品作为一系列的增量结构来设计、编码、集成、测试
- 每个构件由多个相互作用的模块构成，并且能够完成特定的功能

### 优点

- 能在较短时间内向用户提交可完成部分工作的产品
- 逐步增加产品功能可以使用户有较充裕的时间学习和适应新产品，从而减少一个全新的软件可能给客户组织带来的冲击

### 缺点

- 在把每个新的增量构件集成到现有软件体系结构中时，必须不破坏原来已经开发出的产品
- 增量模型本身是自相矛盾的

## 3、可行性研究

### 可行性研究的任务p35

---

- 可行性研究的目的是不是解决问题，而是确定问题是否值得去解决
- 需要进一步分析和澄清问题定义
- 分析员导出系统的逻辑模型，探索可供选择的主要解法

- 对于每种解法，研究技术可行性，经济可行性，操作可行性
- 为每个可行的解法指定一个粗略的实现进度
- 最根本的任务是对以后的行动方针提出建议

## 可行性研究的目标p35

---

用最小的代价在尽可能短的时间内确定问题是否能够解决

## 可行性研究步骤p36

---

- 复查系统规模和目标
- 研究目前正在使用的系统
- 导出新系统的高层逻辑模型
- 进一步定义问题
- 导出和评价供选择的解法
- 推荐行动方针
- 草拟开发计划
- 书写文档提交审查

## 数据流图p40

---

自己看

## 数据字典p47

---

自己看

## p54第五题

---

自己看

## 4、需求分析

### 对系统的综合要求p56（至少五六点）

---

- 功能需求

- 性能需求
- 可靠性和可用性需求
- 出错处理需求
- 接口需求
- 约束
- 逆向需求
- 将来可能提出的要求

## 正式访谈与非正式访谈（展开论述） p58

---

正式访谈时，系统分析员将提出一些事先准备好的具体问题，例如，询问客户公司销售的商品种类、雇佣的销售人员数目以及信息反馈时间应该多快等。

在非正式访谈中，分析员将提出一些用户可以自由回答的开放性问题，以鼓励被访问人员说出自己的想法，例如，询问用户对目前正在使用的系统有哪些不满意的地方。

## 数据模型：实体-联系图（ER图） p62

---

自己看

## 功能模型：数据流图p40

---

自己看

## 行为模型：状态图p65（要会画）

---

自己看

## 区分1:1 1:N M:N p63

---

自己看

## 数据规范化（一、二、三范式） p64

---

- 范式越高，冗余度越小
- 范式越高，表越多，过程越复杂，理解越困难

- 范式越高，数据存储结构与基于问题域的结构间的匹配程度随之下降，在需求变化时，数据稳定性较差
- 范式提高，表访问增多，性能下降
- 另外自己看

## 5、总体设计

### 总体设计目标p91

---

- 概括的说，系统应该如何实现。总体设计又称为概要设计或初步设计。
- 划分出组成系统的物理元素——程序、文件、数据库、人工过程和文档等
- 设计软件结构

### 总体设计过程p92

---

- 设想供选择的方案
- 选取合理的方案
- 推荐最佳方案
- 功能分解
- 设计软件结构
- 设计数据库
- 制定测试计划
- 书写文档
- 审查和复查

### 模块化 抽象 逐步求精p94

---

自己看看

### 耦合p97

---

耦合是对一个软件结构内不同模块之间互连程度的度量

尽量使用数据耦合，少用控制耦合和特征耦合，限制公共环境耦合的范围，完全不用内容耦合。

#### 数据耦合

- 两个模块彼此间通过参数交换信息，且交换的信息仅是数据
- 低耦合，系统至少必须存在

## 控制耦合

- 两个模块彼此间通过参数交换信息，交换的信息中有控制信息（控制信息可能以数据形式出现）
- 中等耦合，增加系统复杂性。往往多余，可分解用数据耦合代替

## 特征耦合

- 把整个数据结构作为参数传递而被调用的模块只需要使用其中一部分数据元素
- 导致数据的访问失去控制，给计算机犯罪提供机会

## 公共环境耦合

- 当两个或多个模块通过一个公共数据环境相互作用时的耦合，公共环境可以是全程变量、共享的通信区、内存的公共覆盖区、任何存储介质上的文件、物理设备等。
- 复杂程度随耦合的模块个数而变化，模块个数增加时复杂程度显著增加
- 若只有两个模块有公共环境，那么这种耦合有下面两种可能：一放一取，这是数据耦合的一种形式，是比较松散的耦合；都放都取，耦合比较紧密，介于数据耦合与控制耦合之间
- 若两个模块共享数据很多，都通过参数传递不方便，可以利用公共环境耦合

## 内容耦合

- 最高程度的耦合，应坚决避免使用
- 一个模块访问另一个模块的内部数据；一个模块不通过正常入口而转到另一个模块的内部；两个模块有一部分程序代码重叠（只可能出现在汇编程序中）；一个模块有多个入口（这意味着一个模块有几种功能）

## 内聚p98

内聚标志着一个模块内各个元素彼此结合的紧密程度，它是信息隐藏和局部化概念的自然拓展。简单地讲，理想内聚的模块只做一件事。设计时应力求做到高内聚，通常中等程度的内聚也可以采用，且效果和高内聚相差不多。

### 偶然内聚 0分 低内聚

- 一个模块完成一组任务，这些任务彼此间即使有关系，关系也是很松散的。

### 逻辑内聚 1分 低内聚

- 一个模块完成的任务在逻辑上属于相同或相似的一类（例如一个模块产生各种类型的全部输出）

### 时间内聚 3分 低内聚

- 一个模块包含的任务必须在同一时间段内执行（例如模块完成各种初始化工作）



## 过程内聚 5分 中内聚

- 一个模块内的处理元素是相关的，而且必须以特定次序执行
- 使用程序流程图作为工具设计软件时，常常通过研究流程图确定模块的划分，这样得到的往往是过程内聚的模块

## 通信内聚 7分 中内聚

- 模块中所有元素都使用同一个输入数据和（或）产生同一个输出数据

## 顺序内聚 9分 高内聚

- 一个模块内的处理元素和同一个功能密切相关，而且这些处理必须顺序执行（通常一个处理元素的输出数据作为下一个处理元素的输入数据）
- 根据数据流图划分模块时，通常得到顺序内聚的模块，这些模块彼此连接往往比较简单

## 过程内聚 10分 高内聚

- 模块内所有处理元素属于一个整体，完成一个单一的功能

## 启发规则（至少五个） p100

---

- 改进软件结构提高模块独立性
- 模块规模应该适中
- 深度、宽度、扇出、扇入都应适当
- 模块的作用域应该在控制领之内
- 力争降低模块接口的复杂程度
- 设计单入口单出口的模块
- 模块功能应该可以预测

## 软件层次图p102

---

自己看

## 6、详细设计

### 详细设计阶段的根本目标p117

---

确定应该怎样具体地实现所要求的系统，经过这个阶段的设计工作，应该得出对目标系统的精确描述，从而在编码阶段可以把这个描述直接翻译成用某种程序设计语言书写的程序

## 设计问题p119

---

自己看

## 判定树 p129 p142第七题

---

自己看

## 程序复杂程度的定量度量p137

---

### McCabe方法

- 会计算环形复杂度
- 会画流图，会转换
- p137下面三个公式
- 自己看

## 7、实现

## 编码风格，好编码p147

---

源程序代码的逻辑简明清晰、易读易懂是好程序的一个重要标准

## 什么叫测试目标p150

---

- 测试是为了发现程序中的错误而执行程序的过程
- 好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案
- 成功的测试是发现了迄今为止尚未发现的错误的测试

## 软件测试准则p150

---

- 所有测试都应该能追溯到用户需求
- 应该远在测试开始前就制定出测试计划
- 把Pareto原理（测试中发现错误的80%很可能是由程序中20%的模块造成的）应用到软件测试中。
- 应该从“小规模”测试开始，并逐步进行“大规模”测试
- 穷举测试是不可能的
- 为了达到最佳的测试效果，应该由独立的第三方从事测试工作

## 测试步骤p151

---

- 模块测试
- 子系统测试
- 系统测试
- 验收测试
- 平行运行

## 测试重点p153

---

- 模块接口
- 局部数据结构
- 重要的执行通路
- 出错处理通路
- 边界条件

## 代码审查p154

---

自己看

## 计算机测试

---

模块并不是一个独立的程序，因此必须为每个单元测试开发驱动软件和（或）存根软件

### 驱动程序

- 一个“主程序”，它接收测试数据，把这些数据传送给被测试的模块，并且印出有关结果

### 存根程序

- 存根程序代替被测试的模块所调用的模块，因此也可以称为“虚拟子程序”。它使用被它代替的模块接口，可能做最少量的数据操作，印出对入口的检验或操作结果，并且把控制归还给调用它的模块

## 集成测试 自顶向下 自底向上p156

---

自己看

## Alpha & Beta 测试p161

---

自己看

## 白盒测试概念p162

---

设计测试方案是测试阶段的关键技术问题。所谓测试方案包括具体的测试目的（例如，预定要测试的具体功能），应该输入的测试数据和预期结果。通常又把测试数据和预期的输出结果称为测试用例。其中最困难的问题是设计测试用的输入数据。

## 逻辑覆盖p162

---

读懂前三种 自己看

## 黑盒测试概念p171

---

黑盒测试着重测试软件功能。不能取代白盒测试，它是与白盒测试互补的测试方法，它很可能发现白盒测试不易发现的其他类型的错误

## 黑盒测试力图发现的错误p171

---

- 功能不正确或遗漏了功能
- 界面错误
- 数据结构错误或外部数据库访问错误
- 性能错误
- 初始化和终止错误

## 等价划分p172

---

等价划分是一种黑盒测试技术，这种技术把程序的输入域划分成若干个数据类，据此导出测试用例。一个理想的测试用例能独自发现一类错误（例如对所有负整数的处理都不正确）

## 边界值分析p175

---

自己看

## 错误推测p175

---

自己看

## 调试概念p176

---

调试（也称为纠错）作为成功测试后的结果出现，也就是说，调试是在测试发现错误之后排除错误的过程。调试是把症状和原因联系起来的尚未被人深入认识的智力过程

## 调试途径p178

---

- 蛮干法
- 回溯法（什么是回溯法，自己看）
- 原因排除法

## 软件可靠性的定义p179

---

是时间段 自己看

## 软件的可用性定义p179

---

是时间点 自己看

## 题型

判断10分 10题

单项选择20分 10题

填空10分 10题

简单15分 5题

综合45分 4题