

# 系分知识点整理 (yhx)

\*一星 ▲三星

## ch01 绪论

- 1) \*信息系统的生命周期：系统规划、分析、设计、实现、运行与评价
- 2) \*信息系统的建设原则：高管介入、用户参与、自顶向下规划、工程化  
(其他原则：创新、整体、发展、经济、先进)

## ch02 经济管理与应用数学

- 1) \*最小生成树：普里姆算法/克鲁斯卡尔算法 (按少取、无回路、线比点少 1、不唯一)
- 2) \*最短路径：少中取少
- 3) \*最大流量： $\Sigma$  (每一条通路减其最小值，直至没有通路)
- 4) 算法
  - a) \*递归：确定初值找出递归关系，如：阶乘、斐波那契数列
  - b) \*分治：分而治之，分解->求解->合并
  - c) \*贪心：不追求最优解，只看局部 (当前)，不看整体 (长远)，不要回溯
  - d) \*动态规划：分治+解决冗余，多阶段拆分成一系列单阶段，然后求解最优
  - e) 迭代法：重复反馈
- 5) \*关键路径：最长路径，不可压缩工期
- 6) \*PERT 期望：服从  $\beta$  分布， $a_i$  乐观时间， $m_i$  最可能时间， $b_i$  悲观时间

$$t_i = \frac{a_i + 4m_i + b_i}{6}, \quad \text{方差} = S^2 = \sigma_i^2 = \frac{(b_i - a_i)^2}{36}, \quad \text{标准差} = S = \sigma$$

标准正态分布概率： $\pm\sigma$  68.3%， $\pm 2\sigma$  95.4%， $\pm 3\sigma$  99.7%

- 7) 均值：

方差：度量与期望值 (均值) 的偏离程度。方差越大，数据波动越大。

标准差：样本方差开平方。衡量离散程度，标准差越小越集中，越大越离散。

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 \quad S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$$

- 8) 线性规划：解二/三元一次不等式，联立+代入

## ch03 操作系统基本原理

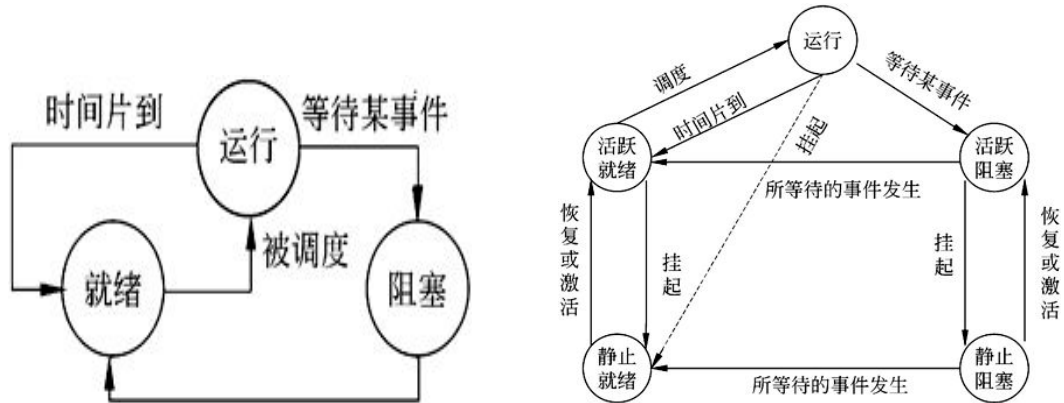
- 1) \*OS

功能：①CPU 管理、②存储器管理、③设备管理、④文件管理、⑤用户接口

类型：单用户、批处理、分时、网络、分布式、嵌入式

- 2) \*分时 OS：CPU 划分时间片轮流给各作业，多路性、独立性、及时性、交互性、同时性

3) \*进程：三态模型 (运行、就绪和阻塞)，五态模型 (运行、活跃阻塞、静止阻塞、静止就绪、活跃就绪)



4) ▲PV 操作

### ch04 数据通信与计算机网络

1) ▲OSI 模型：助记词**巫术忘传会飚鹰**

OSI 模型	TCP/IP 协议					TCP/IP 模型	传输内容	设备
应用层	ftp	telnet	smtp	nfs	snmp	应用层	服务接口	
表示层	[21]	[23]	[25]	[2049]	[161]		统一数据	
会话层							建立会话	
传输层	TCP (面向连接)		UDP (无连接)		传输层	段 segment	流量控制	
网络层	IP	ICMP		ARP	RARP	网际层	报文 packet	路由器
数据链路层	802.3	FDDI	802.5	ARCnet	PPP/SLIP	网络接口层	帧 frame	二层交换机、网桥
物理层	以太	光纤	令牌环	同轴		硬件层	bit	中继器、hub

2) \*帧中继：是一种高性能的广域网技术，运行在 OSI/RM 的物理层和数据链路层，它是一种数据包交换技术，是 x.25 网络的简化版本，比 x.25 网络具有更高的性能和更有效的传输效率。适用：带宽不高（64Kbps~2Mbps）多方通信、长距离、突发性

3) \*网络规划

- ①需求分析：自上而下 SA，功能、通信、性能、可靠性、安全、运维、管理
- ②可研：技术/经济/法律/用户使用
- ③现有网络分析：服务器/客户机数量和位置、并发、日访户数、使用时间、业务数据量、拥塞时间段、协议、通信模式

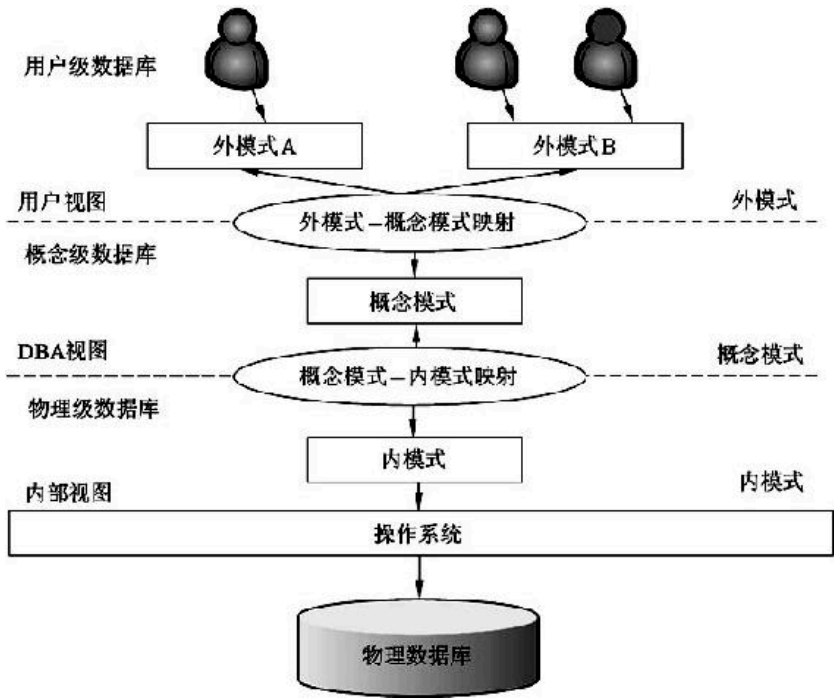
4) 网络设计

- ①任务：总体目标、设计原则、通信与资源子网设计、设备选型、OS、安全
- ②分层设计：核心层、汇聚层、接入层

5) 网络实施：计划、到货、设备安装、系统测试、试运行、培训、系统转换

ch05 数据库

1) \*三级模式两级映射



2) 关系数据库的基本概念

- 关系->表；元组->表中的行；属性（元组的某个分量）->表中的列  
数据库模式->数据库的逻辑设计；数据库实例->给定时刻数据库中数据的一个快照
- a) 超码（键）：一个或多个属性的集合，唯一地标识一个元组
  - b) 候选码（键）：最小超码
  - c) 主码（键）：选定一个候选键为主键
  - d) 外码（键）：非本表的主键，关联表的主键
  - e) 主属性：含候选键的属性为主属性，否则为非主属性

【补充】

逻辑运算符：非 ( $\neg$ )、与 ( $\wedge$ )、或 ( $\vee$ )  
量词：存在 ( $\exists$ ) exist、任意 ( $\forall$ ) all

3) \*基本关系代数运算

运算符		含义	运算符		含义
集合运算符	$\cup$	并差交笛卡儿积	比较运算符	$>$	大于
	$-$			$\geq$	大于等于
	$\cap$			$<$	小于
	$\times$			$\leq$	小于等于
				$=$	等于
				$\neq$	不等于
专门门的运算符	$\sigma$	选择 投影 连接 除	逻辑运算符	$\neg$	非 与 或
	$\pi$			$\wedge$	
	$\bowtie$			$\vee$	
	$\div$				

a) 并 (Union)：交集

- b) 差 (Difference) : 差集
- c) 笛卡尔积 (Cartesian Product)  $\times$ : 相乘
- d) 投影 (Projection)  $\pi$ : 选列, 垂直分割
- f) 选择 (Selection)  $\sigma$ : 选行, 水平分割

#### 4) 扩展关系代数运算

a) 交：交集

b) 自然连接 (Natural Join)  $\bowtie$ ：将关系 R 和 S 中公共属性组满足对应分量相等的元组连接起来，并去掉重复的属性。

A	B	C
a	b	c
b	a	d
c	d	e
d	f	g

(a) 关系 R

A	C	D
a	c	d
d	f	g
b	d	g

(b) 关系 S

A	B	C	D
a	b	c	d
b	a	d	g

$R \bowtie S$

c) 除 (Division)  $\div$

A	B	C	D
a	b	c	d
a	b	e	f
a	b	h	k
b	d	e	f
b	d	d	l
e	k	e	d
e	k	e	f

(a) R

C	D
c	d
e	f

(b) S

A	B
a	b
e	k

(c)  $R \div S$

d) 外连接：左外连接、右外连接、全外连接

A	B	C
a	b	c
b	a	d
c	d	e
d	f	g

(a) 关系 R

B	C	D
b	c	d
d	e	g
f	d	g
d	e	c

(b) 关系 S

A	B	C	D
a	b	c	d
e	d	e	g
e	d	e	c
b	a	d	null
d	f	g	null

(a) 左外连接  $R \bowtie_{\text{L}} S$

A	B	C	D
a	b	c	d
e	d	e	g
e	d	e	c
null	f	d	g

(b) 右外连接  $R \bowtie_{\text{R}} S$

A	B	C	D
a	b	c	d
e	d	e	g
e	d	e	c
b	a	d	null
d	f	g	null
null	f	d	g

(c) 全外连接  $R \bowtie_{\text{F}} S$

#### 4) \*规范化理论：不规范导致数据冗余、增删改异常

1NF：所有属性不可再分

2NF：消除非主属性对候选键的部分依赖

3NF：消除非主属性对候选键的传递依赖

反规范化技术：增加冗余列、增加派生列、重新组表和分割表

#### 5) DB 访问接口：专用调用、ODBC (MS)、JDBC

▲ODBC：优点是统一的方式处理所有的数据库

缺点①只支持关系型数据、②对每种数据库支持不理想、③需要配置

JDBC：纯 java 移植性好，简便，允许使用高级功能

#### 6) \*数据中心建设：6 方面

①构建专用存储系统，集中存储数据

②构建统一运行平台，提高数据处理能力

③建立多种系统应用平台，提高集中运行平台的适应性

④整合优化设备资源，提高集中管理和应用水平

⑤扩展数据备份系统，提高系统可靠性

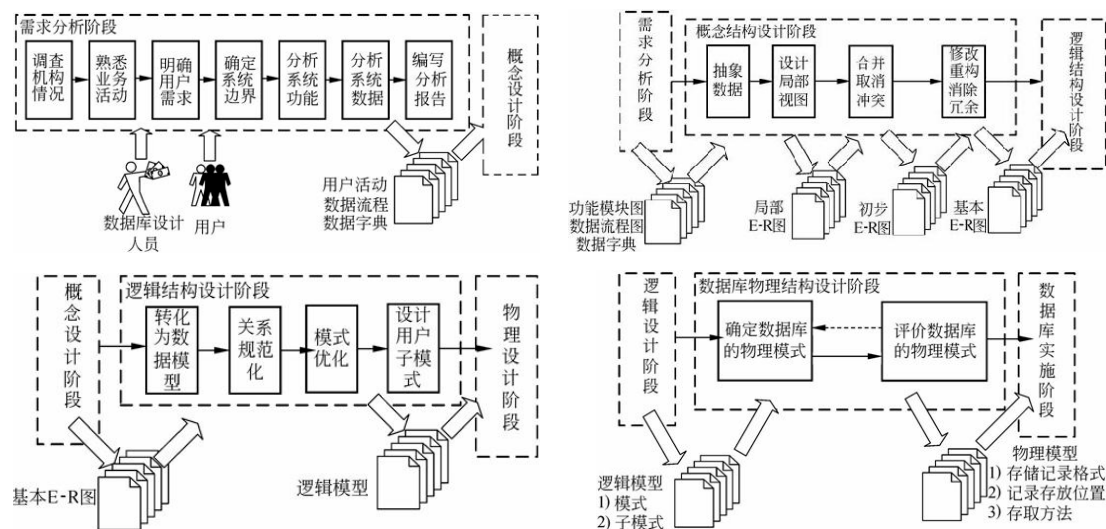
⑥建立集中运行管理机制，实现设备和资源的统一管理

#### 7) ▲DB 设计

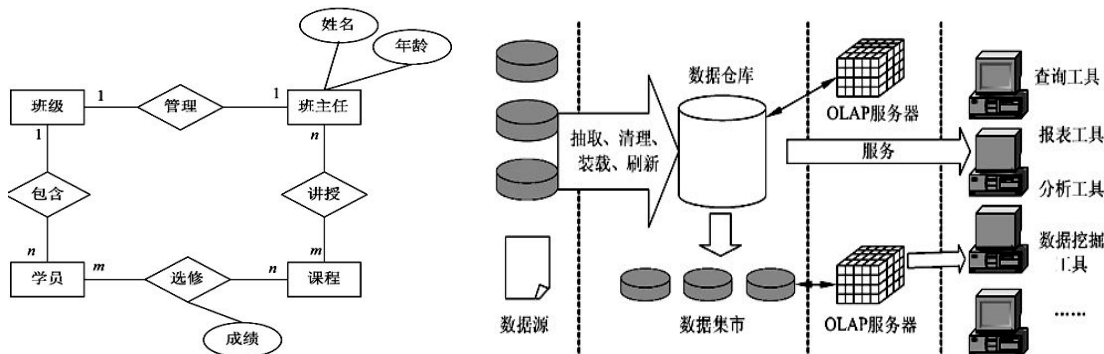
①规划：必要性分析，确定数据库的地位，与其他数据库之间的关系

- ②需求分析：了解用户数据和处理要求，形成 SRS
- ③概念设计：根据需求抽象成不依赖于 DBMS 的概念模型，可邀请用户参与
- ④逻辑(结构)设计：概念模型转化为某 DBMS 的逻辑模型，E-R 图
- ⑤物理(结构)设计：逻辑模型落地到物理结构，如存储结构、存储方法，物理设计的步骤如下：

- (1) 设计存储记录结构，包括记录的组成、数据项的类型和长度，以及逻辑记录到存储记录的映射
- (2) 确定数据存储安排
- (3) 设计访问方法，为存储在物理设备上的数据提供存储和检索的能力
- (4) 进行完整性和安全性的分析与设计
- (5) 数据库程序设计



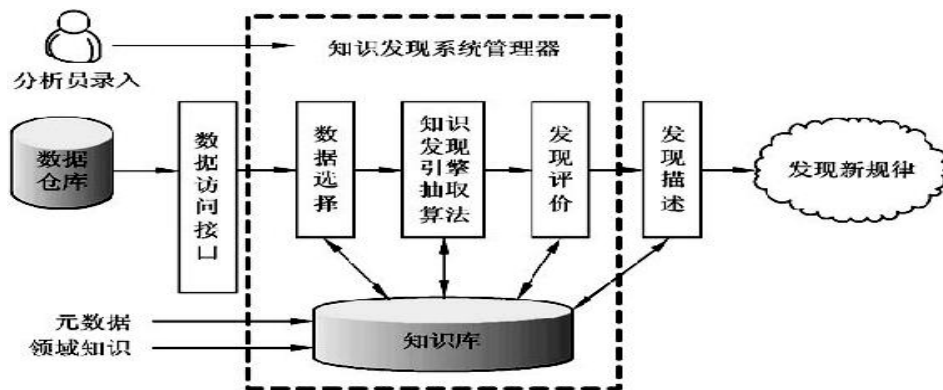
8) \*E-R 图：实体（矩形）、属性（椭圆）、联系（菱形 1:1, 1:n, m:n）



9) \*数据仓库特点：①面向主题、②集成、③相对稳定、④反应历史变化

10) \*ETL：抽取 (Extract)、转换 (Transform)、加载 (Load)

- 11) ▲数据挖掘流程：①问题定义 ②建立挖掘库 ③分析(趋势/规律) ④调整 ⑤模型化(神经网络/决策树/数理统计) ⑥评价解释



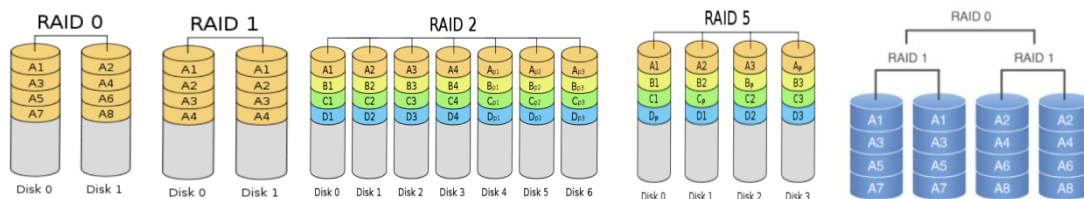
12) 数据挖掘常用技术：决策树、分类、粗糙集、神经网络、关联规则、概念树、遗传算法、依赖性分析、公式发现、统计分析、模糊论、可视化

13) 数据挖掘分析方法：关联分析、序列分析、分类分析、聚类分析、预测、时间序列

## ch06 系统配置与性能评价

1) \*Flynn 分类：SISD、SIMD、MISD(不实际)、MIMD 指令(Instruction)

2) \*RAID：共 8 种 1/2/3/4/5/6/7/10



RAID 0: 无冗余和无校验的数据分块

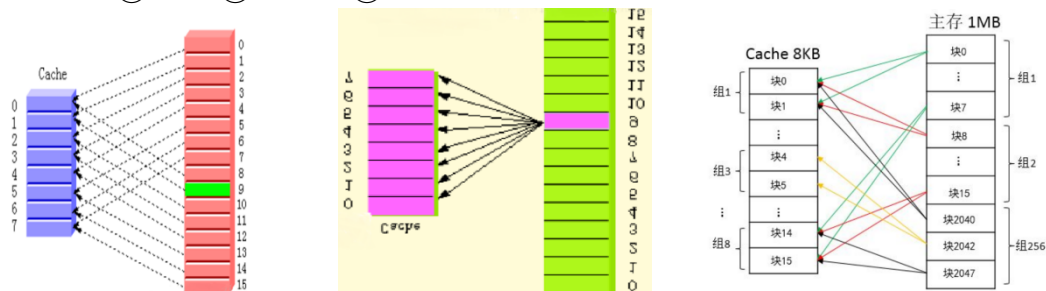
RAID 1: 磁盘镜像阵列, 利用率 50%

RAID 2: 采用纠错海明码的磁盘阵列

RAID 5: 无独立校验盘的奇偶校验码磁盘阵列, 可用容量=(N-1)\*最低容量硬盘容量

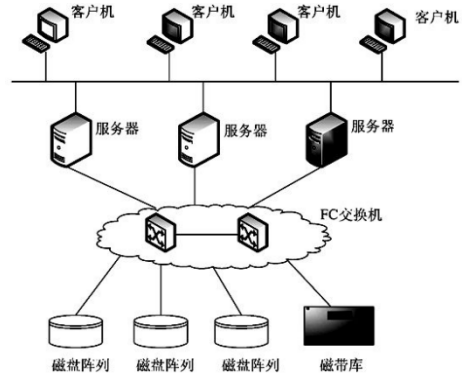
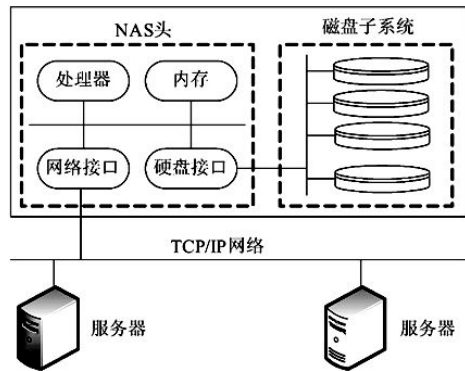
RAID 10: RAID1+0 高可靠性与高性能的组合

3) \*Cache: ①直接 ②全相联 ③组相联



4) \*命中率：设  $H_c$  为 Cache 的命中率， $t_c$  为 Cache 的存取时间， $t_m$  为主存的访问时间，则 Cache 存储器的等效加权平均访问时间  $t_a$  为  $t_a = H_c t_c + (1 - H_c) t_m = t_c + (1 - H_c)(t_m - t_c)$

## 5) \*网络存储



直接附加存储 (Direct Attached Storage, DAS): 分散、直连、直存

网络附加存储 (Network Attached Storage, NAS): 即插即用

存储区域网络 (Storage Area Network, SAN): 高速光纤传输

## 6) \*流水线、时空图

特点: ①连续任务 ②各部件有缓冲寄存器 ③大操作可分解并行工作

理论公式:  $T_k = (t_1 + t_2 + \dots + t_k) + (n-1) * \Delta t$ , 流水线建立分  $k$  段,  $n$  个任务

实践公式:  $T_k = (k+n-1) * \Delta t$

吞吐率:  $TP = n / T_k$

加速比: 顺序执行所用的时间为  $T_0$ , 使用流水线的执行时间为  $T_k$ , 则加速比  $S = T_0 / T_k$

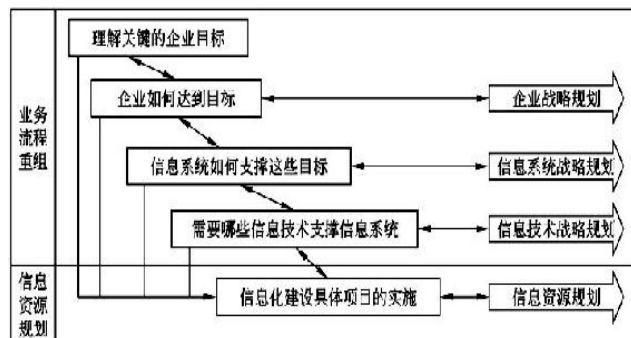
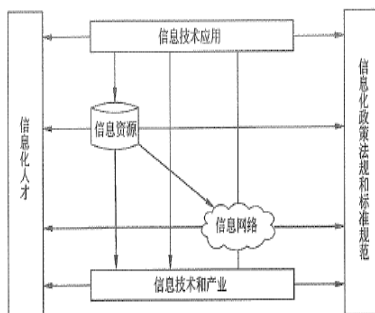
效率:  $E = n$  个任务占用的时空区 /  $k$  个流水段的总时空区  $= T_0 / (k T_k)$

## 7) \*网络性能评估指标

①信道传输速率 ②信道吞吐量和容量 ③信道利用率 ④传输延迟 ⑤响应时间 ⑥负载能力

## ch07 企业信息化战略与实施

1) \*国家信息化体系 6 要素: ①资源 (核心) ②网络 (基础设施) ③技术应用 (龙头) ④产业 (物质基础) ⑤人才 (关键) ⑥政策法规标准 (保障)



2) \*企业信息化方法: ①流程重组 ②核心业务应用 ③信息系统建设 ④主题数据库 ⑤资源管理

## 3) ▲企业信息化规划 (案例题)

依赖关系分析: 涉及业务流程重组、信息技术/系统战略规划、企业战略规划

IT 战略规划步骤: ①分析业务 (现在/未来、当前架构/系统) ②识别机会 ③选择方案

具体内容: ①明确目标和重点 ②成立领导机构 ③需求分析 ④投资预算 ⑤规章制度

## 4) ▲信息系统开发方法

SD 特点: ①目标清晰 ②工作阶段化 ③文档规范化 ④设计方法结构化



SD 缺点: ①周期长 ②难适需求变化 ③很少考虑数据结构

SD 贡献: ①强调开发方法的结构合理性 ②明确划分了规划、分析、设计、实施、维护

OO 思想: 客观世界由各“对象”组成, 对象由属性和操作组成

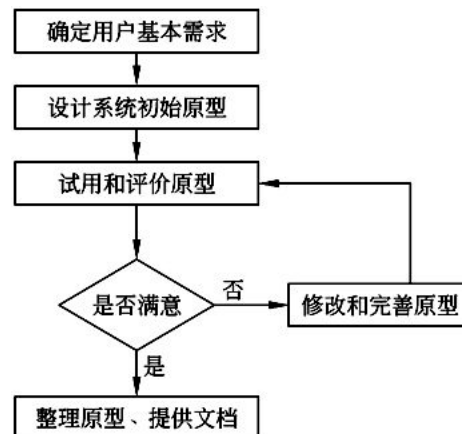
OO 基本概念: 类、对象、继承、封装、消息、多态

SO 思想: 类+对象=构件, 用接口调用, 递进关系: OO->基于构件->SO

SO 抽象级别: 操作(低层)、服务(中层)、业务流程(高层)

服务建模 3 阶段: 服务发现、服务规约和服务实现

原型法开发过程: ①确定需求 ②设计系统初始原型 ③试用和评价 ④修正完善 ⑤整理原型和文档



原型法特点: ①开发周期短 ②以用户为中心 ③利于移交运维

原型法缺点: ①开发环境要求高 ②管理水平要求高

5) 信息系统战略规划 ISSP: BSP、关键成功因素(CSF)、战略集合转化(SST)、战略数据规划(SDP)、信息工程(IE)、战略栅格(SG)、价值链分析(VCA) 战略(Strategic)

6) \*企业系统规划 BSP 原则

- ①支持企业战略目标
- ②表达出企业各管理层的需求
- ③向整个企业提供一致的信息
- ④适应企业组织结构和管理体制的改变
- ⑤由总体信息系统结构中的子系统开始实现

7) ▲BSP 步骤: 定义管理目标、定义管理功能、定义数据分类和定义信息结构

8) \*ERP 开发方法: ①二次开发 ②定制开发, 差异(规划中、实施中、维护中)

从目前 ERP 实施的技术手段来看, 定制开发和二次开发正在相互渗透。套装软件正在提高其开放程度, 开放多种接口, 为企业提供更灵活的二次开发手段。定制开发也正在出现大量经过封装的中间件和应用构件, 大大加速了定制开发和实施的速度。套装软件提出定制化套装软件的概念, 定制开发走产品化的道路, 二者有趋于统一的趋势。

9) ▲BI 阶段: ①数据预处理、②建立数据仓库、③数据分析、④数据展现

实施步骤: ①需求分析 ②仓库建模 ③数据抽取 ④建 BI 分析报表 ⑤用户培训和数据模拟测试 ⑥系统改进完善

10) \*业务流程重组 BPR 性质: ①根本性(why) ②彻底性 ③显著性 ④流程

11) ▲企业集成

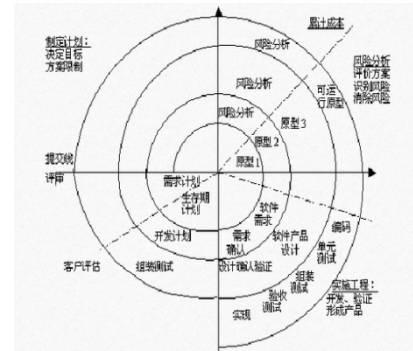
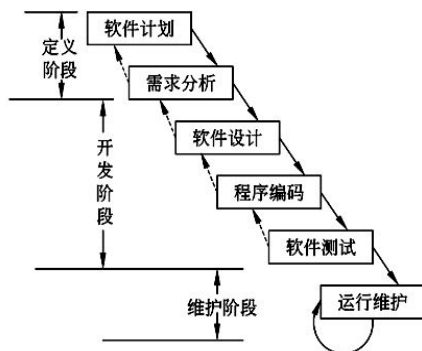
①表示(界面)集成: PC 用户界面、看上去统一实则多系统组成、可能性唯一, 不彻底可能有性能瓶颈

- ②数据集成:白盒, 多源数据、访问公用信息库、多源数据格式不同, 业务逻辑变化
- ③控制(功能/应用):黑盒, 借助远程过程调用, 更灵活性, 更复杂, 业务逻辑不提供API 时难集成
- ④业务流程(过程)
- ⑤企业间应用集成

## ch08 软件工程

1) \*逆向工程:是分析程序, 力图在比源代码更高抽象层次上建立程序的表示过程, 逆向工程是设计的恢复过程。相关概念①重构 ②设计恢复 ③再工程 ④正向工程

2) ▲瀑布模型 6 过程: 软件计划、需求分析、软件设计、程序编码、软件测试和运行维护



优点: 强调开发的阶段性、需求调查和产品设计, 开发人员清楚每步工作, 利于项目管理

缺点: 不能适应需求变化, 反馈不及时, 风险后露, 风控弱, 项目延期, 超预算

3) \*演化模型: 不完整定义需求, 快速原型->不断改进->最终产品

优点: 功能一经开发就测试, 提早验证, 利于高质量

缺点: 用户接触未稳定功能, 对开发人员和用户造成负面影响

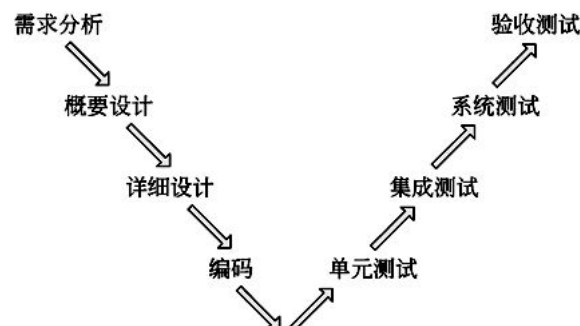
4) ▲螺旋模型 4 特点: 螺旋模型是瀑布模型与演化模型相结合, 每次迭代都包括**制订计划**、**风险分析**、**实施工程**和**客户评估** 4 个方面的工作, **强调风险分析**

5) ▲喷泉模型: 以用户需求为动力, 以对象为驱动力的模型, 自下而上相互重叠、多次反复, 就像水喷上去又可以落下来, 似一个喷泉

6) ▲变换模型: 它对形式化的软件规格说明进行一系列自动或半自动的程序变换, 最后映射为计算机能够接受的软件系统, 必要时可以修改软件需求直到原型被确认为止。

7) \*智能模型: 综合了上述若干模型, 并把专家系统结合在一起。该模型应用基于规则的系统, 采用规约和推理机制。

8) ▲V 模型: 助记词**延续膝盖, 吉祥单编**



延(验收) 续(需求) 膝(系统) 盖(概要), 吉(集成) 祥(详细) 单(单元) 编(编码)

## ch09 系统规划

### 1) ▲系统规划步骤

- (1) 对现有系统进行初步调查
- (2) 分析和确定系统目标
- (3) 分析子系统的组成和基本功能
- (4) 拟定系统的实施方案
- (5) 进行系统的可行性研究
- (6) 制订系统建设方案

### 2) ▲可行性研究：①经济 ②技术(技术/资源/目标) ③法律 ④用户使用(管理、运行)

3) \*复利：P 元在 n 年后的价值， $F = P \times (1+i)^n$  法  $P = F / (1+i)^n$ ，分母为折/贴现系数

4) \*净现值 NPV：在生命周期内各年的净现金流量按照一定的、相同的折现率折现到初时的

现值之和，即 
$$NPV = \sum_{t=0}^n \frac{(CI - CO)_t}{(1+i)^t}$$
，其中  $(CI - CO)_t$  为第 t 年的净现金流量，CI 为现金流入，CO 为现金流出，i 为折现率。

### 5) \*投资回收期、投资回报率

$T = \text{累计净现金流量开始出现正值的年份数} - 1 + \text{上年累计净现金流量} / \text{当年净现金流量}$

$T_p = \text{累计折现值开始出现正值的年份数} - 1 + \text{上年累计折现值} / \text{当年折现值}$

投资回收期 =  $1 / \text{动态投资回收期} \times 100\%$

## ch10 系统分析

### 1) \*详细调查方法

- ①收集资料 ②开调查会 ③个别访谈 ④书面调差 ⑤抽样调查 ⑥现场观摩 ⑦业务实践 ⑧阅读历史文档

## ch11 软件需求工程

1) \*需求层次：①业务需求(范围) ②用户需求(做什么) ③系统需求(功能、性能、约束)

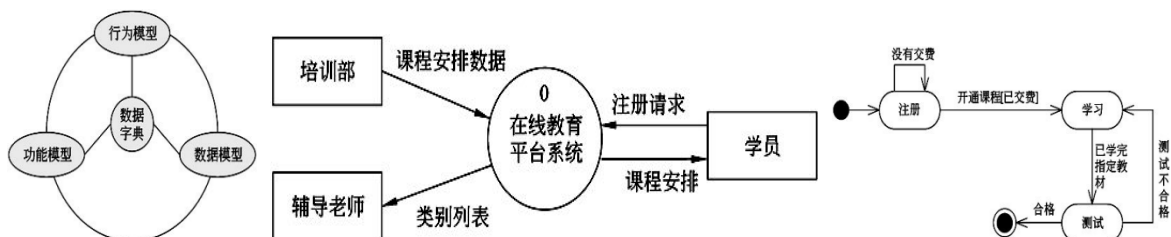
2) \*质量功能部署 QFD：常规、期望、意外(兴奋)

3) \*需求获取方法：①用户访谈 ②问卷调查 ③采样 ④情节串联板 ⑤联合需求计划

样本大小 =  $0.25 * (\text{可信度系数} / \text{可接受的错误})^2$

4) 需求分析方法：SA、OOA(找出类和对象)

▲SA：自顶而下，逐层分解，核心是数据字典，3 个层次模型(数据 ER、功能 DFD、行为/状态 STD)



### 5) \*UML 的 3 个部分

- ①构造块(事物/关系/图)

②规则(命名/范围/可见性/完整性/执行)

③公共机制(说明、修饰、公共分类[类与对象、接口与实现]、扩展机制[约束、构造、标记值])

▲4种关系: ①依赖 ②关联 ③泛化(继承反关系) ④实现

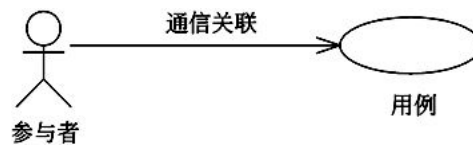
\*5种视图: ①逻辑(设计) ②进程(并发同步) ③实现(开发) ④部署(物理) ⑤用例(需求)

\*14种图:

静态: ①类 ②对象 ③构件 ④组合结构 ⑤用例 ⑥部署 ⑦包 ⑧制品

动态: ①顺序(序列) ②通信(协作) ③定时 ④状态 ⑤活动 ⑥交互概览

构建用例模型: 识别参与者、合并需求获得用例、细化用例描述和调整用例模型



6) \*需求验证: 也称为需求确认, 其活动是为了确定以下几个方面的内容:

(1) SRS 正确地描述了预期的、**满足**项目干系人需求的系统行为和特征。

(2) SRS 中的软件需求是从系统需求、业务规格和其他来源中**正确**推导而来的。

(3) 需求是**完整**的和高质量的。

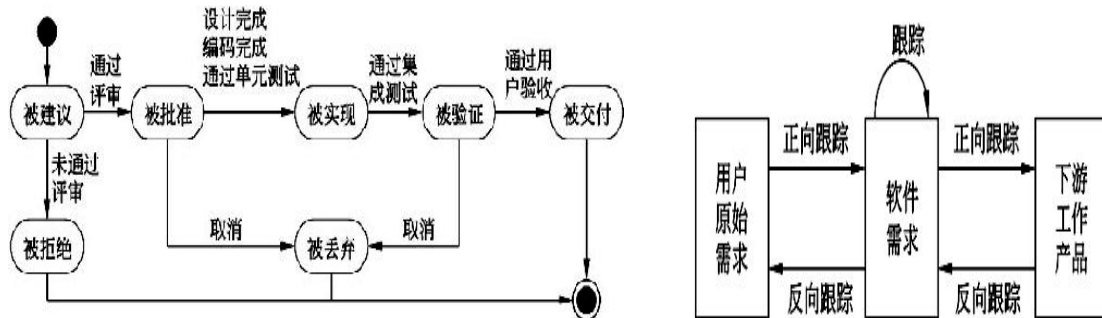
(4) 需求的表示在所有地方都是一**致**的。

(5) 需求为继续进行系统设计、实现和测试提供了足够的**基础**。

7) 需求评审要求: (分层次、正式与非正式、分阶段、精选评审者、培训、检查单、评审流程、跟踪、充分准备)

注意: 在整个需求开发的过程中, 需求获取、需求分析、需求定义、需求验证 4 个阶段不是瀑布式的发展, 而是应该采用迭代式的演化过程。

8) \*需求变更管理(需求基线、需求状态、需求变更)



8) \*需求风险管理

①无足够用户参与 ②忽略用户分类 ③需求不断增 ④模棱两可 ⑤不必要特性 ⑥SRS 过于精简 ⑦不准确估算

## ch12 软件架构设计

1) ▲经典架构: 助记词**输掉钩机仓** 输-数据流, 掉-调用, 钩-构件, 机-虚拟机, 舱-仓库

①**数据流**风格: 批处理序列、管道/过滤器两种风格

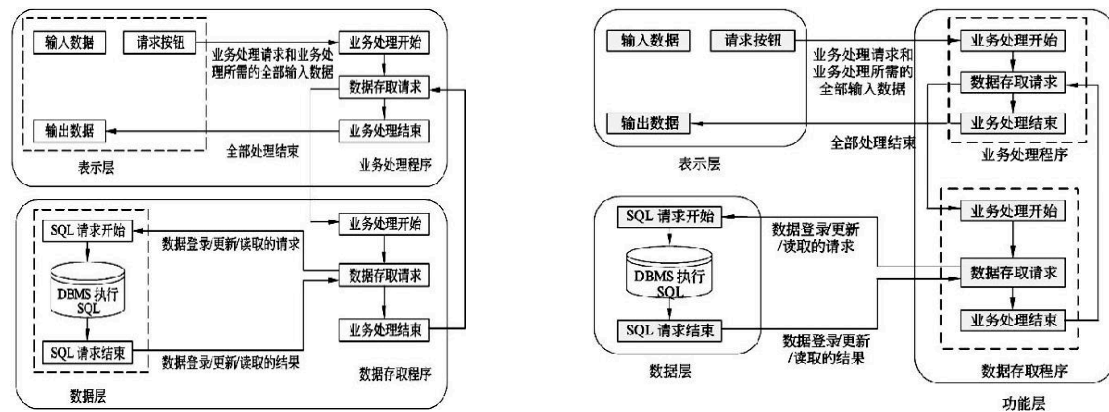
②**调用/返回**风格: 主程序/子程序、数据抽象和面向对象, 以及层次结构

③**独立构件**风格: 进程通信、事件驱动的系统

④**虚拟机**风格: 解释器、基于规则的系统

⑤**仓库**风格: 数据库系统、黑板系统和超文本系统

## 2) ▲层次架构



### ▲二层架构缺点:

- (1) 开发成本较高
- (2) 客户端程序设计复杂
- (3) 用户界面风格不一，使用繁杂，不利于推广使用
- (4) 软件移植困难
- (5) 软件维护和升级困难
- (6) 新技术不能轻易应用
- (7) 可扩展性差
- (8) 系统安全性难以保证

三层架构：表示层、功能层、数据层

### ▲三层优点:

- (1) 允许合理地划分三层的功能，使之在逻辑上保持相对独立性，从而使整个系统的逻辑结构更为清晰，能提高系统的可维护性和可扩展性。
- (2) 允许更灵活、有效地选用相应的平台和硬件系统，使之在处理负荷能力上与处理特性上分别适应于结构清晰的三层，并且这些平台和各个组成部分可以具有良好的可升级性和开放性。
- (3) 系统的各层可以并行开发，各层也可以选择各自最适合的开发语言，使之能并行且高效地进行开发，达到较高的性能价格比。对每一层的处理逻辑的开发和维护也会更容易些。
- (4) 利用功能层可以有效地隔离表示层与数据层，未授权的用户难以绕过功能层而利用数据库工具或黑客手段去非法地访问数据层，这就为严格的安全管理奠定了坚实的基础。

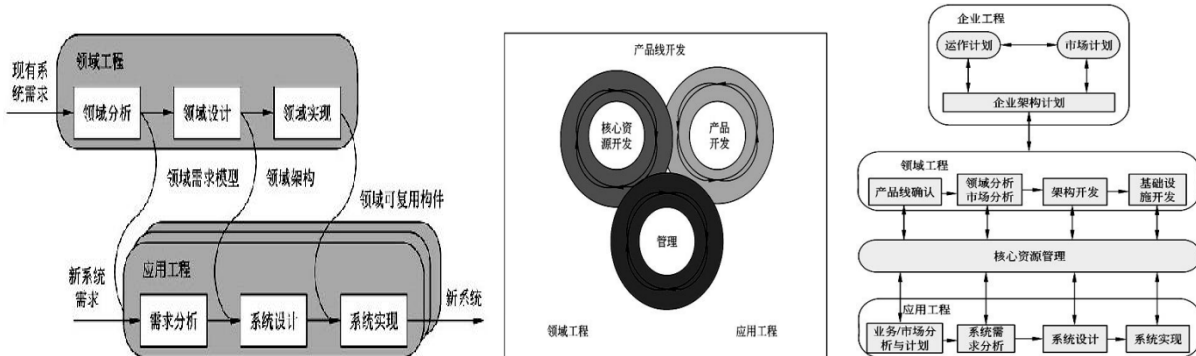
## 3) \*SOA 定义

- (1) W3C 的定义：SOA 是一种应用程序架构，在这种架构中，所有功能都定义为独立的服务，这些服务带有定义明确的可调接口，能够以定义好的顺序调用这些服务来形成业务流程。
- (2) Service-architecture.com 的定义：服务是精确定义、封装完善、独立于其他服务所处环境和状态的函数。SOA 本质上是服务的集合，服务之间彼此通信，这种通信可能是简单的数据传送，也可能是两个或更多的服务协调进行某些活动。服务之间需要某些方法进行连接。
- (3) Gartner 的定义：SOA 是一种 C/S 架构的软件设计方法，应用由服务和使用者组成，SOA 与大多数通用的 C/S 架构模型不同之处，在于它着重强调构件的松散耦合，并使用独立的标准接口。

#### 4) \*SOA 关键技术

- ①UDDI: 统一描述、发现和集成, 包含数据模型、API、注册服务
- ②WSDL: web 服务描述语言, 基于 xml 的语法定义, 包含服务实现定义、服务接口定义
- ③SOAP: 封装、编码规则、RPC 表示、绑定。SOAP 消息含封装 Envelop, 头 Header, 体 Body
- ④REST: 事物皆资源, 资源 ID, 操作资源接口, 操作不影响 ID, 所有操作无状态。

#### 5) \*产品线过程模型: ①双生命周期 ②SEI ③三生命周期



#### 6) 产品线的建立方式: 演化、替代、新演化、新开发

- ①现有产品演化为产品线
- ②用软件产品线替代现有产品集
- ③全新软件产品线的演化
- ④全新软件产品线的开发

### ch13 系统设计

#### 1) \*SD: 面向数据流, 以 SRS/DFD/数据字典为基础, 自顶向下、逐步求精、模块化

#### 2) \*内聚、耦合

耦合类型	描述
非直接耦合	两模块无直接关系, 联系完全通过主模块的控制和调用
数据耦合	借助参数表传递简单数据
标记耦合	通过参数表传递记录信息 (数据结构)
控制耦合	传递的信息中包含用于控制模块内部逻辑的信息
外部耦合	访问同一全局变量 (非全局数据结构), 不是通过参数表传递
公共耦合	访问同一个公共数据环境 (如全局数据结构、共享通信去、公共内存)
内容耦合	不通过正常入口直接访问另模块的内部数据, 代码重叠, 模块有多个入口

内聚类型	描述
功能内聚	完成单一功能, 各部分协同工作, 缺一不可
顺序内聚	处理元素相关, 且顺序执行
通信内聚	所有处理元素集中在一个数据结构的区域上
过程内聚	处理元素相关, 必须按特定的次序执行
瞬时内聚	任务必须在同一时间间隔内执行
逻辑内聚	完成逻辑上相关的一组任务
偶然内聚	完成一组没有关系或松散关系的任务



高内聚低耦合：耦合低使得模块间尽可能相对独立，从而各模块可以单独开发和维护，可以提高软件质量；内聚高使得模块的可理解性和维护性大大增强。

3) \*OOD 思想：OOD 是 OOA 方法的延续，其基本思想包括**抽象**、**封装**和**可扩展性**，可扩展性主要通过**继承**和**多态**来实现。在 OOD 中，数据结构和在数据结构上定义的操作算法封装在一个对象之中。由于现实世界中的事物都可以抽象出对象的集合，所以 OOD 方法是一种更接近现实世界、更自然的系统设计方法。

类：①实体类 ②边界类 ③控制类

4) OOD 原则：

- ①开闭原则：开放扩展，关闭修改
- ②里氏替换原则：基类可替换成子类，反之不行
- ③依赖倒置原则：抽象不依赖于细节，细节依赖于抽象
- ④组合/聚合复用：用组合/聚合 (has-a 耦合低灵活)，少用继承 (is-a 破坏封装性)
- ⑤接口隔离原则：一个接口完成一个角色，尽量提供小接口，不要提供大的总接口
- ⑥最小知识原则迪米特法则 (Law of Demeter)：信息隐蔽

## 5) \*设计模式

创建型	用于创建对象	工厂方法模式 (宝马厂)、 <b>抽象工厂</b> 模式 (系列)、 <b>单例</b> 模式 (打印机)、 <b>建造者</b> 模式 (造房子)、 <b>原型</b> 模式 (拷贝)。5【助记词： <b>单抽元件厂</b> 】
结构型	处理类或对象的组合	<b>适配器</b> 模式 (电源)、 <b>装饰器</b> 模式 (变形金刚)、代理模式、 <b>外观</b> 模式 (界面)、桥接模式 (图案+色彩)、组合模式 (盘子+水果)、 <b>享元</b> 模式 (黑白棋)。7【助记词： <b>外侨组员戴饰佩</b> 】
行为型	描述类与对象怎样交互、怎样分配职责	策略模式 (不同交通工具)、 <b>模板</b> 方法模式 (取号+办业务：存、取、转)、观察者模式 (MVC，红绿灯)、 <b>迭代</b> 子模式 (换频道)、责任链模式 (报销签字)、命令模式 (遥控器)、备忘录模式 (Redo/Undo)、状态模式 (订房)、 <b>访问者</b> 模式 (划价/药房)、 <b>中介</b> 者模式、解释器模式。11 观摩 <b>对策</b> ，【助记词：观摩对策，责令解放， <b>戒忘</b> 台】

## ch14 系统实现与测试

### 1) 自动化测试

\*优点：①↑速度 ②↑效率 ③准确性 ④连续运行 ⑤模拟现实

▲缺点：不是万能的，测试人员在决定使用自动化测试，必须考虑到其受约束的地方，例如，自动化测试不能取代手工测试，能够发现的缺陷不如手工测试；自动化测试对所测产品质量的依赖性大，不能提高有效性，可能会制约软件开发，以及测试工具本身不具备想象力等。

\*自动化测试工具

- ①单元测试工具：JUnit
- ②负载和性能测试：LoadRunner, QALoad, SQA Load, Visual Quality
- ③GUI 功能测试：WinRunner, QARun, SQA Robot, Visual Test Suite
- ④基于 Web 应用的测试：E-TestSuite, WorkBench, WAS Tool, Link Sleuth

### 2) 测试方法

\*静态：桌前检查、审查、走查、静态分析 (控制流、数据流、接口、表达式)

\*动态 (白盒)：覆盖 (语句、判定、条件、条件/判定、条件组合、修正条件/判定、路径)

\*动态(黑盒)：功能分解、等价类划分、边界值分析、判定表、因果图、状态图、随机测试、错误推测、正交试验法

3) \*测试类型：GB/T 15532—2008，软件测试可分为单元测试、集成测试、配置项测试、系统测试、验收测试和回归测试等

4) ▲集成测试策略

①基于分解的集成策略：非渐增式(一步到位)、渐增式(单元+集成、混合增量三明治)

②基于功能的集成策略：确定优先级、分析功能路径、增加关键功能

③基于调用图的集成策略：有向图(节点：程序模块；边：程序调用)

5) 其他测试类型：配置项测试(对照 SRS)、验收测试(SRS)、确定测试(内部确认， $\alpha/\beta$ )、回归测试

6) \*测试过程：策划、设计、执行、总结

7) \*测试组织：GB/T 15532—2008

测试类型	组织
单元测试	开发方、供方，或委托第三方
集成测试	供方，测试人员与开发人员独立，或委托第三方
配置项测试	供方，实施人员，开发人员配合，或国家认可的第三方测试机构
系统测试	供需方，实施人员，开发人员配合，或国家认可的第三方测试机构
验收测试	实施人员，或国家认可的第三方测试机构
回归测试	与其所对应软件测试类别的组织管理相同或相似

## ch15 系统运行与维护

1) ▲遗留系统特点

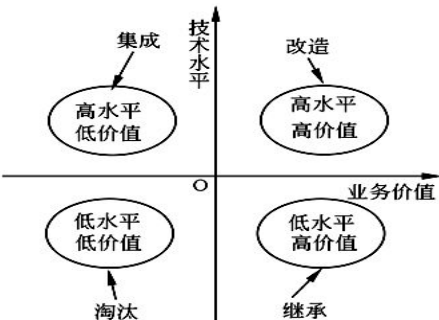
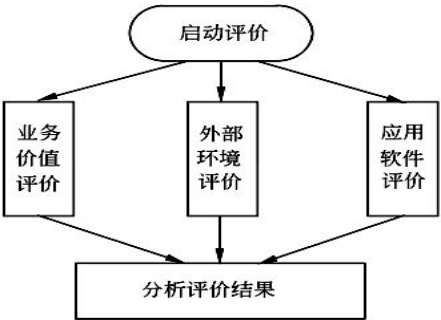
(1) 系统虽然完成企业中许多重要的业务管理工作，但仍然不能完全满足要求。一般实现业务处理电子化及部分企业管理功能，很少涉及经营决策。

(2) 系统在性能上已经落后，采用的技术已经过时。例如，多采用主机/终端形式或小型机系统，软件使用汇编语言或第三代程序设计语言的早期版本开发，使用文件系统而不是数据库。

(3) 通常是大型的软件系统，已经融入企业的业务运作和决策管理机制之中，维护工作十分困难。

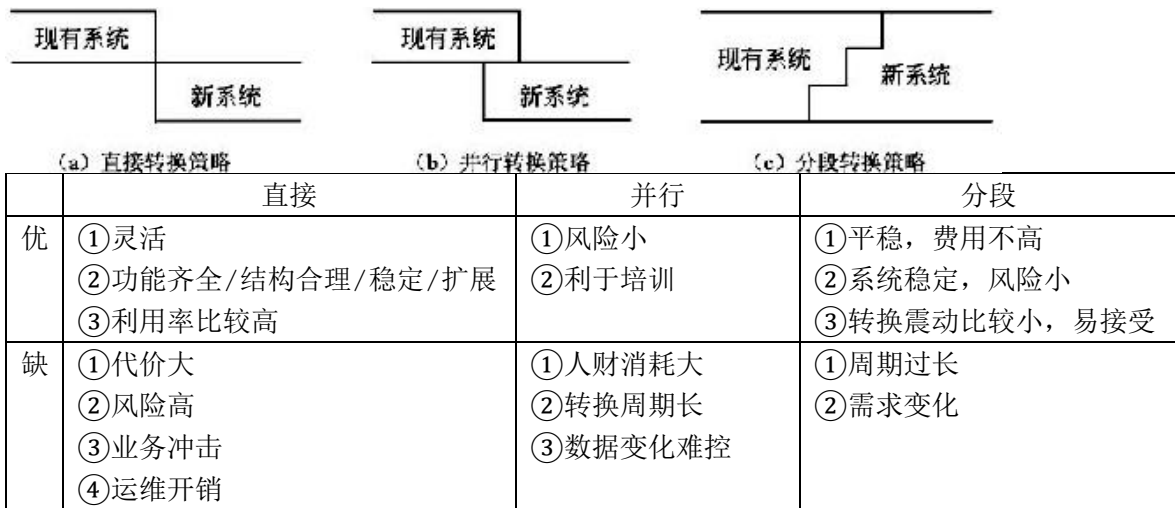
(4) 没有使用现代信息系统建设方法进行管理和开发，现在基本上已经没有任何文档，很难理解。

2) 演化策略：①淘汰 ②集成 ③改造 ④继承



3) ▲新旧系统转换





### 3) ▲数据迁移准备工作

- (1) 待迁移数据源的详细说明，包括数据的存放方式、数据量和数据的时间跨度。
- (2) 建立新旧系统数据库的数据字典，对现有系统的历史数据进行质量分析，以及新旧系统数据结构的差异分析。
- (3) 新旧系统代码数据的差异分析。
- (4) 建立新旧系统数据库表的映射关系，对无法映射字段的处理方法。
- (5) 开发或购买、部署 ETL 工具。
- (6) 编写数据转换的测试计划和校验程序。
- (7) 制定数据转换的应急措施。

4) \*软件维护：助记词**就是鱼丸** ①**纠正性** ②**适应性** ③**完善性** ④**预防性**

5) \*工程监理：“四控、三管、一协调”，即投资控制、进度控制、质量控制、变更控制，安全管理、信息管理、合同管理和沟通协调。

## ch16 新技术应用

### 1) \*中间件功能

- ①负责客户机与服务器之间的连接和通信，以及客户机与应用层之间的高效率通信机制。
- ②提供应用层不同服务之间的互操作机制，以及应用层与数据库之间的连接和控制机制。
- ③提供一个多层架构的应用开发和运行的平台，以及一个应用开发框架，支持模块化的应用开发。
- ④屏蔽硬件、操作系统、网络和数据库的差异。
- ⑤提供应用的负载均衡和高可用性、安全机制与管理功能，以及交易管理机制，保证交易的一致性。
- ⑥提供一组通用的服务去执行不同的功能，避免重复的工作和使应用之间可以协作。

### 2) \*中间件

主要的中间件：远程过程调用 (RPC)、对象请求代理 (Corba)、远程方法调用 (RMI)、面向消息的中间件

底层中间件：java 虚拟机、公共语言运行库、自适应通信环境

通用型中间件：RPC、ORB、面向消息中间件等

集成型中间件：WorkFlow、EAI 等

## ch17 嵌入式

1) \*特点: ①专用 ②实时 ③软硬依赖 ④处理器专用 ⑤多技术 ⑥透明(只管用) ⑦资源受限

## ch18 系统安全性分析与设计

- 1) \*系统安全分类: ①实体安全 ②运行安全 ③信息安全 ④人员安全
- 2) 安全体系结构: ①物理层(机房) ②系统层(OS 漏洞) ③网络层 ④应用层 ⑤安全管理
- 3) 安全保护 5 等级: 《计算机信息系统安全保护等级划分准则》(GB17859-1999) 规定
- I 用户自主保护级: 隔离用户与数据, 普通内联网
- II 系统审计保护级: 粒度更细的自主访问控制, 内联网或国际网的商务活动、非重要单位
- III 安全标记保护级: 各级国家机关、金融机构、邮电通信、能源与水源、交通
- IV 结构化保护级: 中央级、广视、重要物资储备、应急部门、尖端科技企业、国防
- V 访问验证保护级: 国防关键部门
- 4) \*对称加密: ①DES 钥 56 位 ②3DES 钥 112 位 ③IDEA 钥 128 位 ④RC5 ⑤AES
- \*非对称加密: ①RSA 秘钥 512 位 ②Elgamal ③背包算法 ④Rabin ⑤D-H
- ⑥ECC(椭圆曲线)
- 对称速度快, 用于大批量; 非对称速度慢, 用于数字签名
- 信息摘要: ①SHA-1 160 位摘要 ②MD5
- 数字证书包含: CA 的信息、用户信息、用户公钥、CA 签发时间和有效期
- 【例】用户乙收到甲数字签名后的消息 M, 为验证消息的真实性, 首先需要从 CA 获取用户甲的数字证书, 该数字证书中包含(甲的公钥), 并利用(CA 的公钥)验证该证书的真伪, 然后利用(甲的公钥)验证 M 的真实性。
- 5) \*防火墙功能: ①访问控制 ②内容控制 ③日志 ④集中管理 ⑤自身安全 ⑥流控/NAT/VPN
- ▲局限性: ①限制关闭服务 ②内部攻击 ③SLIP/PPP 进入 internet ④不透明 ⑤无法防病毒
- 6) 系统安全保障系统内容:
- ①建立统一的身份认证体系
  - ②建立统一的安全管理体系
  - ③建立规范的安全保密体系
  - ④建立完善的网络边界防护体系
  - ⑤安全立法、组织管理、行政管理和业务组织、标准和评估、技术保障

## ch19 系统可靠性分析与设计

1) \*可靠性指标

平均无故障时间 MTTF

平均故障修复时间  $MTTR = 1/\mu$  修复率  $\mu$  是指单位时间内可修复系统的平均次数

平均故障间隔时间  $MTBF = MTTR + MTTF$ , 一般 MTTR 很小, 所以通常认为  $MTBF \approx MTTF$

系统可用性  $= MTTF / (MTTR + MTTF) \times 100\% = MTTF / MTBF \times 100\%$

2) ▲冗余

### ①结构冗余

静态/屏蔽/被动：通过冗余表决和比较来屏蔽系统中出现的错误，如三/多模

动态/主动：通过故障检测、故障定位、故障恢复到容错目的，多重模块待机

混合：静态+动态，取两者之长，成本高，仅用在可靠性要求极高的场合

### ②信息冗余：增加一些信息来保证运行结果，如检错码、纠错码

### ③时间冗余：降低运行速度为代价，重复执行比较结果(相同=故障,不同=故障)

### ④冗余附加：指为实现上述冗余技术所需的资源和技术，包括程序、指令、数据，以及存放和调用它们的空间

2) \*双机容错：①热备(一主一备) ②互备 ③双工(负载均衡)

3) 可靠度：串联  $R = R_1 * R_2 * \dots * R_n$  并联  $R = 1 - (1 - R_1) * (1 - R_2) * \dots * (1 - R_n)$

## ch20 项目管理

### 1) \*开发计划内容

①工作计划 ②人员 ③设备采购和资源供应 ④配置管理 ⑤进度 ⑥成本投资

⑦质量保证 ⑧风险管理 ⑨文档编制 ⑩支持

### 2) \*范围管理 6 过程

①规划范围管理 ②收集需求 ③定义范围 ④创建 WBS ⑤范围确认 ⑥范围控制

WBS 中最底层工作单元为工作包，工作包非常具体

范围确认并不是确认要干什么，而是确认可交付成果是否满足项目干系人的要求

### 3) 成本管理 4 过程

①规划成本 ②估算成本 ③指定预算 ④控制成本

PV: plan value 要干的活 EV: end value 干完的活 AC: acture cost 实际花费

进度偏差  $SV = EV - PV$

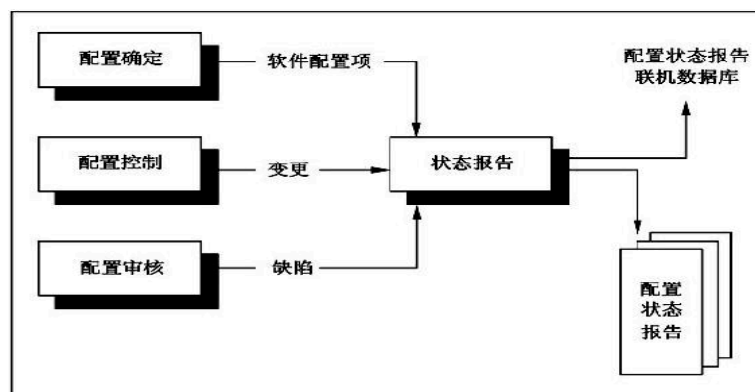
进度绩效  $SPI = EV / PV$

成本偏差  $CV = EV - AC$

成本绩效  $CPI = EV / AC$

4) \*管理储备：不是项目成本基线(挣值分析)的一部分，但包含在预算中，未知的未知，没花要还给业主，如果要动用需经业主同意，要预留不能扣。

### 5) ▲配置管理 6 过程：①制定计划 ②配置标识 ③配置控制 ④配置状态报告 ⑤配置审计 ⑥发布管理和交付



软件版本：草稿 0.YZ，正式 X.Y，修改 X.YZ，开发库、受控库、产品库

### 6) 软件质量 6 特性：助记词功能靠用小壶移

GB/T16260.1-2006：①功能性 ②可靠性 ③易用性 ④效率 ⑤维护性 ⑥可移植性

老 7：因果图 流程图 核查表 排列/帕累托图 直方图 控制图 散点图(刘英只点劣质茶)

新 7: 亲和图 过程决策程序图 PDPC 相互关系/关联图 树形图 优先矩阵图 活动网络图 矩阵图 (矩树相亲策动优)

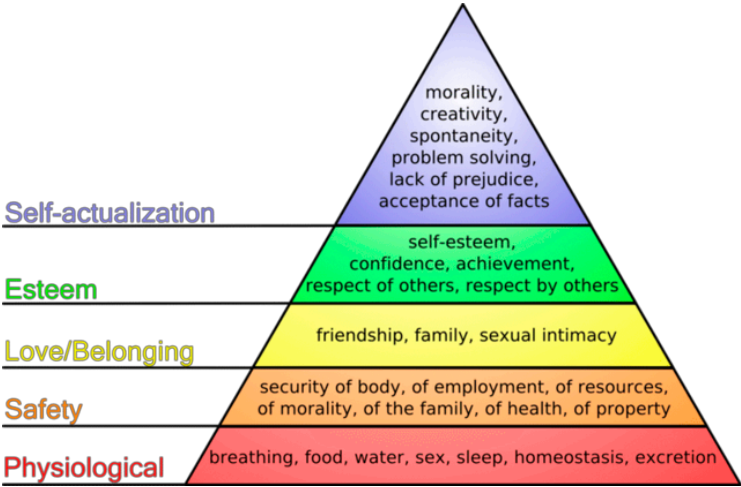
7) 责任分配矩阵 (Responsibility Assignment Matrix, RAM)

	项目干系人				
	A	B	C	D	E
单元测试	S	A	I	I	R
集成测试	S	P	A	I	R
系统测试	S	P	A	I	R
验收测试	S	P	I	A	R

负责人 A 参与者 P 输入 I 评审 R 确认者 S

解决冲突 6 种方法: ①问题解决 ②合作 ③强制 ④妥协 ⑤ 求同存异 ⑥撤退

8) 马斯洛需要层次理论



生理上的需要、安全的需要、社交的需要、尊重的需要和自我实现的需要

9) ▲风险管理 6 过程: ①计划编制 ②识别 ③定性分析 ④定量分析 ⑤应对 ⑥监控

案例 (▲)

①②开发/设计/建模/测试 ③嵌入式 ④数据库 ⑤Web

	试题一 (必答)	试题二	试题三	试题四	试题五
18	系统分析	系统分析设计	嵌入式安全性设计	数据库设计 (视图)	web 架构设计 (REST)
17	基于微服务的系统开发	系统数据分析与建模	嵌入式多核程序设计技术	数据库分析与建模	web 系统架构设计
16	软件系统可行性分析	系统设计	嵌入式系统能耗管理设计	数据库设计	web 应用开发框架
15	用例测试	系统开发	嵌入式系统开发	银行中间业务系统开发	汽车配件网上交易系统
14	需求建模	ERP 系统规划	嵌入式实时操作系统	系统运行与维护	web 应用

## 论文 (▲)

①测试 ②开发方法 ③风险管理 ④需求分析

	试题一	试题二	试题三	试题四
18	论信息系统 <b>开发方法</b> 论	论软件 <b>构件</b> 管理及其应用	论软件系统 <b>需求</b> 获取技术及应用	论数据挖掘方法及应用
17	论 <b>需求分析</b> 方法及应用	论企业应用集成	论数据流图在系统分析与设计中的应用	论软件的系统测试及其应用
16	论软件 <b>需求验证</b> 方法及其应用	论软件的系统测试及其应用	论软件开发模型及应用	论信息系统规划及实践
15	论项目 <b>风险管理</b> 及其应用	论软件系统测试及其应用	论软件系统的容灾与恢复	论非关系型数据库技术及应用
14	论信息系统 <b>开发方法</b> 及应用	论业务流程建模方法及应用	论数据库集群技术及应用	论企业信息 <b>集成</b> 技术及应用

## 其他

### 1)、多媒体

**表示媒体**是为了加工、处理和传输感觉媒体而人为研究、构造出来的一种媒体，它有各种编码方式，如：文本编码、图像编码和声音编码等

**表现媒体**是指进行信息输入和输出的媒体，如：键盘、鼠标、扫描仪、话筒和摄像机等输入媒体以及显示器、打印机和扬声器等输出媒体。

声音

图像

视频

### 2)、原码、反码、补码

正数：三码一致

负数：反码=[符号位不变，后边按位取反]；补码=[符号位不变，反码末位加 1]

### 3) 校验码

海明码校验

CRC 校验

### 4) 病毒

“熊猫烧香”是一种经过多次变种的计算机蠕虫病毒

“红色代码”病毒是一种新型网络病毒，其传播所使用的技术可以充分体现网络时代网络安全与病毒的巧妙结合，将网络蠕虫、计算机病毒、木马程序合为一体，开创了网络病毒传播的新路，可称之为划时代的病毒。

“冰河”木马病毒是国人编写的一种黑客性质的病毒，感染该病毒之后，黑客就可以通过网络远程控制该电脑。主要用于远程监控。

“爱虫”病毒，是一种蠕虫病毒，这个病毒可以改写本地及网络硬盘上面的某些文件。用户机器染毒以后，邮件系统将会变慢，并可能导致整个网络系统崩溃。

### 5) 敏捷方法

极限编程 (XP)：敏捷开发的典型方法之一，是一种轻量级（敏捷）、高效，低风险、柔性、可预测的、科学的软件开发方法，它由价值观、原则、实践和行为 4 个部分组成。其中 4 大价值观为沟通、简单性、反馈和勇气。

水晶法 (Crystal)：水晶方法体系与 XP 一样，都有以人为中心的理念，但在实践上有所不同。水晶方法体系考虑到人们一般很难严格遵循一个纪律约束很强的过程，认为每一种不同的项目都需要一套不同的策略、约定和方法论。因此，与 XP 的高度纪律性不同，水晶方法体系探索了用最少纪律约束而仍能成功的方法，从而在产出效率与易于运作上达到一种平衡。也就是说，虽然水晶系列不如 XP 那样的产出效率，但会有更多的人能够接受并遵循它。

并列争球法 (Scrum)：用迭代的方法，其中把每 30 天一次的迭代称为一个“冲刺”，并按需求的优先级来实现产品。多个自组织和自治小组并行地递增实现产品。协调是通过简短的日常会议来进行的。

自适应软件开发 (ASD)：ASD 的核心是三个非线性的、重迭的开发阶段：猜测，合作与学习。