

1 计算机组成

1.1 计算机组成与分类

1.1.1 计算机的组成

6 部件：

控制器，是分析和执行指令的部件

运算器，用于逻辑运算也叫算术逻辑单元 ALU

内存储器，用于存储运算中间结果

外存储器

输入输出设备

（控制器与运算器在一起叫做中央处理器 CPU）

1.1.2 计算机的分类

Flynn 分类法

1.单指令流单数据流（ SISD）

2.单指令流多数据流（ SIMD）

3.多指令流单数据流（ MISD）

4.多指令流多数据流（ MIMD），SMP 对称处理与 MPP 海量并行处理结构均属于 MIMD

其他分类方法：冯氏分类法， Handler 分类法， Kuck 分类法

1.2 多级存储器体系

存储器种类：

顺序存取，按顺序进行存取操作，磁带存储器

直接存取，根据唯一地址标识直接找到存储块进行操作，磁盘存储器

随机存取，通过地址随机存取，主存储器（内存）采用该方式

相联存取，通过内容随机存取， Cache 缓存采用该方式

1.2.1 主存储器（内存）

- 1.RAM 随机存取存储器，可写可读断电不保存数据，RAM 又分为 DRAM(动态 RAM，随时间推移而消失)和 SRAM(静态 RAM，断电前一直保存)，容量小价格高
- 2.ROM 只读存储器，一般用于 BIOS 的存储
- 3.PROM 可编程 ROM，只能写入一次
- 4.EPROM 可擦除的 PROM，需要紫外线照射 15-20 分钟可擦除
- 5.E2PROM 电可擦除 EPROM
- 6.闪存存储区，闪存如 U 盘
- 7.CAM 相联存储器，基于内存进行访问，如 cache

1.2.2 辅助存储器（外存磁盘如硬盘）

1.磁带存储器

软盘

2.磁盘存储器

就是分区的圆柱体同心圆，每个面上有磁道与扇区，有多个盘面组成的柱性磁盘组

磁盘总容量 = 盘面数 * 每面磁道数 * 每道扇区数 * 每个扇区字节数

磁盘的数据传输速率 = 磁道上字节数 / 每转 1 圈时间

3.RAID 存储，磁盘冗余阵列，分为 8 个级别分别为：

RAID0,无冗余无校验

RAID1，磁盘镜像阵列

RAID2，采用纠错海明码的磁盘阵列

RAID(3,4)，采用独立校验盘，进行奇偶校验码的磁盘阵列

RAID5，分布式奇偶校验码的磁盘阵列

RAID6，独立数据硬盘，与双独立分布式校验方案

RAID7，高异步 I/O，高速传输阵列

RAID10，综合多个 RAID 等级组成，目前该等级被广泛使用

4.光盘存储器，CD-ROM 压缩盘，DVD 数字视频光盘等

1.2.3 Cache 缓存

1.Cache 原理

CPU 需要的数据先从缓存中找，如果有则直接调用（访问命中），如果没找到再到内存中读取，读取完再送回 CPU 与 Cache 中

系统平均周期 = 缓存周期 * 缓存命中率 + 内存周期 * (1 - 缓存命中率) // 由于缓存中未命中的概率就是在内存中取值的概率

多少位就是 2 的几次幂，例如 32 位系统可以支持 2 的 32 次幂长度的数据也就是 4G 内存，64 位操作系统可支持更高内存

1Byte=8bits

1 汉字 =2B (字节) =16b

1024B=1KB

1024KB=1MB

1024MB=1GB

1024GB=1TB

1024TB=1PB

2.映射机制

存储容量与存储地址的关系：容量的大小转换为 2 的 n 次幂，则地址就用 n 位表示

如 $128 \times 4096B = 2$ 的 19 次幂，就是需要 19 位的内存地址

直接映射：Cache 将主存中的信息地址映射到 Cache 中，主存与 Cache 分成容量相同的块，然后将一个主存的块映射到 Cache 的特定位置上（特定位置是指，将主存地址拆分后四位为 Cache 块内地址，中间 10 位是 Cache 块号）

全相联映射：主存的页对应 Cache 的页，淘汰可替换

组相联映射：块大小相同，组数相同

3.淘汰算法

先进先出 FIFL

最近最少使用淘汰法 LRU

4.写操作

写直达，写入 Cache 时同时写入内存

写回，写入 Cache 时先不写入内存，等在缓存中失效时再写入内存

标记法，标记在 Cache 中修改的信息，读取的时候通过标识位判断

1.3 输入输出接口

1.3.1 输入输出方式

1.程序控制方式，通过指令控制 I/O，但是程序需要时刻查询 I/O 设备的状态，是否完成

2.程序中断方式，CPU 不必去监控 I/O 设备完成情况，I/O 设备完成数据传输后发出中断信号通知 CPU，CPU 在去处理

3.DMA 工作方式，DMAC(DMA 控制器)，CPU 与 DMAC 共享总线，DMA 时 CPU 暂时放弃系统总线控制交给 DMAC 控制

4.通道方式，采用通道程序

5.输入输出处理机，专用处理机，用于大型高效计算机

1.3.2 总线和接口

总线的定义：是一组进行互连和传输信息（指令、数据、地址）的信号线，是计算机内部各个部件链接的桥梁

1.总线分类

按位置分：CPU内部的总线叫内部总线，CPU与内存的总线叫外部总线

按总线功能分：地址总线（传地址）、数据总线（传数据）、控制总线（传控制信号）

按总线在系统中的位置分：机内总线（IDE 链接硬盘的、SCSI小型计算机系统接口）、机外总线（USB可用于链接外部设备）

按用途分：局部总线、系统总线、通信总线

2.总线的标准 :IEEE

3.接口分类：串行接口（一次传 1 位信息）、并行接口（一次传送多位信息）

4.常见接口：

ESDI(加强型小型设备接口)

IDE(磁盘接口)

SCSI(大容量存储设备、光驱接口)

PCMCIA(笔记本内存卡接口)

IEEE-1394数码相机、摄像机接口标准)

USB串行总线式接口，USB1.0速度 12Mbps,USB2.0 速度 480Mbps,USB3.0 速度 4.8Gbps

1.4 各种体系结构

1.复杂指令系统计算机(CISC):指令多、各个指令使用频率悬殊、长度不固定

2.精简指令系统计算机(RISC):指令少、寻址方式少、长度固定，采用Cache方案提高指令获取速度

1.4.2 流水线技术

1.参数计算：

1 个任务分成 n 个子任务，每个子任务需要时间 t ，则完成这个任务需要 nt 时间，

k 个任务顺序执行需要 knt 时间

k 个任务使用流水线则需要：

$$nt + (k-1)t = (n+k-1)t$$

如果分解的子任务用时不同则 t 取用时最长的子任务时间：

$$(t_1 + t_2 + \dots + t_n) + (k-1)t_{\max}$$

用时 = 完成一个任务的时间 + 剩余任务数 $(k-1)$ * 最长子任务时间

吞吐率：单位时间内完成的任务数（秒 s） $1s=10^9$ 次 ns

加速比：不适用流水线用时 / 流水线用时

2. 影响流水线运行的因素

转移指令，流水线无法重叠执行破坏流水线机理

共享资源访问冲突，子任务间存在资源冲突，会破坏流水线

响应中断，执行一半中断

3. 非线性流水线

并行流水线：

公式与流水线一样就是 任务数 / 并行流水线的数量，因为同时执行因此相当于执行如： 3 条

并行流水线同时执行，假设每条流水线任务数相同，则只用执行 $1/3$ 的任务的时间

1.4.3 并行处理

1. 超级标量处理机

2. 超级流水线处理机

3. 超长指令字处理机

4. 向量处理机

5. 多处理机系统

6. 大规模并行处理机，MPP

7. 对称多处理机，SMP

1.4.4 互联网络

2. 操作系统

2.1 操作系统的类型与结构

2.1.1 操作系统类型

1. 批处理

2. 分时

3. 实时

- 4.网络
- 5.分布式

操作系统基本功能：
处理机管理 /进程管理
存储管理
设备管理
文件管理
作业管理

2.1.2 操作系统结构

- 1.无序结构，模块化，模块间通过接口调用
- 2.层次结构，分多层，层次见单向依赖
- 3.面向对象，基于面向对象思想
- 4.对称多处理，多处理机共享内存
- 5.微内核，微内核就是抽象层，将操作系统的功能抽象出来基础功能，一般用于嵌入式操作系统

cpu-寄存器 -一级缓存 -二级缓存 /三级缓存 -内存 -二级存储（外存，外部存储如硬盘）

2.2 处理器管理

2.2.1 进程状态

进程是动态概念，程序是静态概念，进程是程序的动态运行，需要各种资源如 CPU,内存等
进程由程序、数据、进程控制块（PCB）组成

1.三态模型

运行，得到资源并进行计算
就绪，得到资源，等待执行
等待，等待资源或人工干预

2.五态模型

静止就绪，就绪队列中已有就绪任务，多余的就绪任务短期不能调用，放到二级存储中
静止阻塞，阻塞队列中已有阻塞任务，多余的短期不能调用，放到二级存储中
活跃就绪，内存中没有活跃就绪状态的任务时，到静止就绪中获取
活跃阻塞，内存中没有活跃阻塞状态的任务时，到静止阻塞中获取
运行，得到时间片，满足资源要求，在 CPU中处理

状态切换的目的是协调系统资源，由于系统资源有限因此需要通过优先级或条件进行协调资

源，使系统正常运行

挂起状态的进程一般被转换到二级存储中（外存），减少内存损耗

静止阻塞 - 活跃阻塞：系统腾出资源空间并且优先级高的先被激活

挂起的进程不参与调度

必须被激活后才能调入内存中等待执行

挂起的进程只能由操作系统或父进程激活

2.2.2 信号量与 PV 操作

$P(S)$ $S-1$ $S<0$ 挂起

$V(S)$ $S+1$ $S\leq 0$ 触发 $P(S)$ 操作

1. 互斥 $S=1$ PV成对出现

2. 同步 $S=0$ PV多个进程中分别出现

3. 生产消费

空闲区的空闲数量 $S1=N$

已填充空闲区数量 $S2=0$

互斥 $S3=1$

图 2-3

存货的要先判断仓库中有没有空位

如果仓库有空位就执行 $P(S1)$ 空位减少一个

如果没有空位则 $S1<0$ 存货的挂起等待

存完货后要通知取货的库中已经有一个货物了

提货的要先判断仓库里有没有货

如果没有货则挂起等待

如果有货则先 $P(S2)$ 将货物提取一个货物

提取后执行 $V(S1)$ 通知存货的已经有一个空位

图 2-4

不能同时放也不能同时读取因此要用互斥

2.2.3 死锁问题

1. 死锁发生的必要条件

互斥条件

保持与等待条件

不可抢占条件

循环等待条件

2. 银行家算法

表 2-2

申请的总资源不能多于系统资源数

可分期请求资源，请求总数不超过最大需求量

可推迟分配等待资源

申请后的剩余资源能不满足其他进程的总申请资源的需求，
如果可以满足还要测试能满足完成条件的进程完成后释放的资源是否可以满足其他进程全部需求

3.解决死锁的策略

死锁预防：破坏死锁发生条件，如一次性申请全部资源

死锁避免：采用银行家算法

死锁检测：检测是否发生死锁，发生则使用死锁接触策略处理

死锁解除：剥夺发生死锁的进程，强制回收

2.2.4 管程与线程

管程就是将临界资源的调用封装起来，
需要调用临界资源的进程需要进入管程操作，
而管程一次只能进入一个进程进行操作达到进程互斥的目的

线程：

进程申请的资源可以被线程共享使用

一个进程创建时默认创建一个线程也叫主线程，
一个进程可以创建多个线程实现多项任务并行执行，多个线程共享进程资源。

线程解决了进程的并行处理以及进程创建子进程的资源开销

2.3 文件管理

2.3.2 文件的物理组织

连续文件：记录存储在硬盘上的连续的物理块中，通过起始位置及文件长度可获得文件

优点：存取速度快

缺点：由于需要确定长度，因此不适合经常修改的情况

串联文件：物理块中存储下一个物理块的指针，形成一个类似链表的结构

优点：充分利用存储空间

缺点：速度慢不适用于随机存取，只使用顺序访问

索引文件：将物理块的地址通过索引表管理起来

优点：适合随机存取，也适应顺序存取

缺点：索引表占用磁盘空间，而且需要
2 次访问磁盘（ 1 次读取索引表，一次读取物理块内容）

解决方法：将索引表读入内存

UNIX 使用的是三级索引结构

2.3.3 树形目录结构

树形目录，绝对路径（唯一），相对路径（根据工作目录确定），工作目录（当前目录）

NTFS与 FAT32

NTFS快，空间大，分区可达到 2T，碎片少，安全性高

FAT32: 分区最大可达 32G

2.3.4 存储空间管理

存储空间管理：就是管理空闲的物理块

1.空闲表法：将空闲的盘块及数量记录在空闲区的空闲表项中

2.空闲链表法：

将空闲盘块拉成一条链表，通过增加删除操作管理空闲块

链表中的盘区上增加下一个空闲盘区的指针

3.位图法：将空闲盘块通过二进制方式 0 空闲 1 已分配来管理空闲盘区，可将位图放到内存。

4：成组链接法：结合空闲表和空闲链表法，适用于大型文件

2.4 存储管理

2.4.1 地址变换

存储地址，取决于存储空间的大小，二进制或 2 的几次方，如 2 的 32 次方就是 32 位的，64 次方的就是 64 位

地址变换：用硬盘顶替部分内存的工作，使用户可以使用比内存地址大的地址空间来编程，因此用户使用的逻辑地址要转换为内存地址（实际地址）

转换方法：

静态重定位：在程序执行前进行地址映射，无需硬件支持，无法实现虚拟存储

动态重定位：在访问时进行地址转换，支持虚拟存储

存储最小单位是 bit (b) ,字节是 Byte(B),1B=8b

2.4.2 存储组织

常见虚存组织有段式，页式，段页式（最常用）

段页式组织解决逻辑地址转物理地址的方式； 段内页号对应的地址： 高位是页号， 低位是偏移

2.4.3 存储管理

对虚拟存储器的管理设计到的问题及策略

载入（调入）：1 现用现调； 2 提前预调

放置：放到内存分区

置换：内存满了，需要将不常用的页置换到外存虚拟存储中腾出地方给要使用的页

1.置换算法：

最优，淘汰将来不经常使用的，该算法较难实现

随机，容易误杀

先进先出

最近最少使用

2.局部性原理：

可能出现打击率的事件如下：

刚访问的还能再访问；

访问某位置的还会访问该位置附近的

2.5 作业管理

2.5.1 作业的状态

作业创建时会建立一个作业控制块 JCB, 用于管理控制储存作业的信息

作业生命周期：

1.提交：从输入设备提交到外存，传输信息到外存

2.后备：信息传输全部传输到外存后，建立作业控制块

3.执行状态：就是进程的 5 态模型 -从外存调度进入内存执行

4.完成：作业执行完毕，资源还未回收时的状态

2.5.2 作业调度

作业调度：

- 1.高级调度：作业调度，用于在后备作业队列中选择一个或一组一批作业，为他们建立进程分配资源，然后执行
- 2.中级调度：交换调度，用于内外存间的调度，即挂起与激活的调度
- 3.低级调度：进程调度就是调度就绪的进程

作业调度算法：

- 1.先来先服务：不利于短作业
- 2.短作业优先：不利于长作业
- 3.响应比高优先：响应比 $=1+\text{等待时间} / \text{执行时间}$ ，此算法需要计算耗用资源
- 4.优先级调度：优先级高先执行

2.6 设备管理

设备管理程序的功能：

- 1.提供进程调用设备接口
- 2.设备分配给进程
- 3.设备与 cpu 并行操作
- 4.缓冲区管理，解决设备与 CPU内存处理速度不对称问题

2.6.1 磁盘调度算法

磁盘访问流程：寻道，等待扇区旋转，数据传输；速度取决于寻道时间

- 1.FCFS先来先服务，时间长
- 2.SSTF, 最短寻道有限，容易出现饥饿现象又叫“饥饿”
- 3.scan,磁头移动方向上最短时间
- 4.N-SCAN, 分步 scan，扫描期间只接受等待服务的请求
- 5.C-SCAN循环扫描，磁头单向移动的 N-SCAN算法

2.6.2 虚拟设备与 SPOOLing技术

虚拟设备，就是实现多用户多进程访问输入输出设备

原理：通过缓冲区缓冲多个请求到队列中，通过模拟输入输出并来缓冲输入输出进程

优点：实现虚拟设备，多台虚拟设备共用共享一个物理设备，每个进程不会感觉到共享设备，而是独享设备

2.7 网络操作系统

2.8 分布式操作系统

3.数据库系统

3.1 数据库管理系统 DBMS

数据库系统 DBS包括 4 个组成部分：数据库、硬件、软件、数据库管理员（ DBA）

数据库管理方式的特点：

- 1.复杂的数据模型来标识数据结构
- 2.与应用程序的独立性，物理和逻辑上的独立
- 3.提供用户访问接口
- 4.并发控制、恢复、完整性和安全性
- 5.灵活性

3.2 数据库模式

1.三级模式

外模式 -用户级，用户视图

概念模式 -DBA 视图

内模式 -物理模式，数据库文件的物理存储

二级映射

1.外模式 -概念模式映射

2.概念模式 -内模式映射

二级独立

1.物理独立 -用户应用于物理存储独立，各自更改不影响对方

2.逻辑独立 -用户应用独立与逻辑存储，各自修改不相互影响

三个抽象级别

用户级，用户视图，对应与外模式

概念级， DBA 视图，对应概念模式

物理级，内部视图，对应与内模式

3.3 数据模型

3.3.1 关系模型分类

概念模型（用于数据库设计）包括：

E-R实体-联系模型

基本数据模型（用于数据库建模）包括：

层次模型

网状模型

关系模型

面向对象模型

基本数据模型组成

- 1.数据结构 -系统静态特性
- 2.数据操作 -系统动态特性
- 3.完整性约束 -完整性规则

3.3.2 关系模型

关系模型基本概念：

- 1.域就是属性的取值范围
- 2.属性就是一列名称
- 3.元组就是一条数据
- 4.关系就是表与表之间的关联关系
- 5.主属性就是主键
- 6.非主属性就是非主键
- 7.分量就是一个元组中的一列属性值

- 1.关系型数据库的模型

```
DB={STUDENT(name,age,work,sex,birthday),  
    TEACHER(name,age,work)  
}
```

- 2.关系

表与表之间的关系

3.3.3 规范化理论

- 1.存储异常问题：

数据冗余，拆表，冗余的数据放到另外一张表中，进行关联

修改异常，一堆多关系设置，不要把动态信息冗余到各条记录后面，防止共同属性修改导致需要修改几百条引用的信息

插入异常，拆表问题

删除异常，拆表问题，冗余数据在一起，导致删除某条数据冗余的数据同时被删除因为是一条记录的不同属性

2.函数依赖

3.范式

概念：

主属性，关键属性，可唯一标识一个元组的属性组（{学号 +姓名 +性别}，{身份证号 +姓名 +性别 +年龄}）

候选码，可唯一标识一个元组的 1 个主属性（如身份证号，学号）

范式：

第一范式（1NF），各个属性不可分割

第二范式（2NF），如果非主属性完全依赖候选码并且是 1NF

第三范式（3NF），满足 1NF，非主属性不依赖候选码

BC 范式（BCNF），满足 1NF，全部属性都不依赖于候选码

3.4 数据操作

3.4.1 集合运算

交集， $R \cap S$ ，共同的元素

并集， $R \cup S$ ，全部元素

差， $R - S$ （ R 中有但 S 中没有的元素）， $S - R$ （ S 中有 R 中没有的元素）

笛卡尔积（ X ）， $R \times S$ ，集合乘法例如 $\{a,b,c\} \times \{1,2\} = \{a1,a2,b1,b2,c1,c2\}$ ，笛卡尔积后的元组数量 = R 的数量 * S 的数量

3.4.2 关系运算

1.投影，取若干列

2.选择，根据条件筛选元组

3.自然连接，根据相同的键值进行关联

4.左右外链接，以哪个为主，为主要的要全属性

5.完全外链接，2方为主的链接

3.4.3 元组演算

3.4.4 数据库访问接口

1.专用接口

2.ODBC

应用程序、驱动管理、驱动、数据源

3.JDBC

java 数据库连接

3.5 数据库的控制功能

事务 ACID特性

A:原子性，要么全做要么全不做

C:一致性，数据库状态一致性

I:隔离性，事务间互不干扰

D:持久性，事务提交后将永久改变数据，即将数据写入磁盘

事务关键事件：

开始事务 begin transaction

提交事务 commit

回滚事务 ROLLBACK

3.5.1 并发控制

1.并发操作的问题：

丢失修改，同事读取，不同时提交，先提交的被后提交的覆盖。

不可重复读，读取后有事务修改了该数据，导致再次读取时值不同

读脏数据，修改后撤销，导致修改后撤销前读取的值成为脏数据

2.封锁的类型

解决并发问题的方法：

排它锁（X锁），只能加 X锁的事务读取及修改该数据

共享锁（S锁），可共同读取但均不可修改

3.封锁协议：

一级协议，加 X锁

二级协议，一级协议前加 S锁

三级协议，一级协议前加 S 锁至事务结束

两段锁，扩展阶段是申请锁，加锁前申请锁，收缩阶段，释放锁，释放期间不允许申请锁，容易发生死锁

颗粒度：

颗粒度小并发性高开销大

颗粒度大并发性低开销小

4.死锁：

资源附加多种锁，然后事务互相等待对方解锁导致死锁。

解决方法：

预防法，顺序申请，将申请的事务排队编号，逐个申请

解除法，需要 2 个程序，一个检测死锁程序，一个解锁程序

3.5.2 备份与恢复技术

引发系统故障原因大致如下：

事务内部的故障

系统范围内的故障，系统问题引起的故障

介质故障，硬件故障

计算机病毒

1.数据备份

冷备份：拷贝数据库文件到磁盘，缺点需要数据库停止使用后可以拷贝

热备份：通过语句导出数据，缺点导出过程中有修改会导致备份的数据与库中的数据不一致

增量备份：每次只备份修改的部分，加入到之前备份的数据中

2.日志文件

数据库运行过程中的全部操作都记录在日志中，包括数据备份时也叫加入日志，防止备份过程中的修改，通过日志可以知道备份库与现库的差异

先修改日志原则：修改数据前要先存到日志中后再修改数据库

3.数据恢复

redo 重做，undo 撤销或未做完

1) 事务故障的恢复：此过程是系统自动完成的

步骤：

反向扫描日志文件，查找事务的操作

对事务的更新进行逆操作

继续反向扫描日志文件，继续查找事务操作继续执行上述过程

直至读到此事务的开始标记为止

2) 系统故障的恢复：在系统重启时自动完成的

正向扫描日志文件，找出故障发生前已经提交的事务标记入重做队列，未完成的事务标记入撤销队列

对撤销队列中的事务进行撤销操作，反向扫描日志文件将 UNDO 事务的更新操作执行逆操作

对重做队列中的各个事务进行重做处理，正向扫描日志文件对每个 redo 事务执行日志文件登记的操作

3) 介质故障与病毒破坏的恢复

恢复近期数据库备份脚本

从故障点开始反向扫描日志，将已提交的事务标记入 redo 队列

从其实点开始正向扫描日志，将 redo 的事务重做，并恢复至故障前某一时刻的状态

4) 有检查点的恢复技术

建立检查点，每个检查点都记录当前状态的全部事务清单

3.5.3 数据库的安全性

1. 用户认证

口令认证，用户名密码

强身份认证，如证书，智能卡，指纹等

2. 用户角色

一般 3 类角色

1. 数据库登陆角色，只能查

2. 数据库资源管理，可以修改数据

3. DBA，可以创建用户创建表，授权与削权

3. 数据授权

库级别：可访问授权库内的全部内容

表级别：可访问授权的表

行级别：某表的某些行

属性级别：某些属性

静态授权：自己创建的数据自带权限

动态授权：管理员给予的权限

4 数据库视图

视图的功能：

限制到行

限制到列

多表关联

统计信息，如 count (*) 等

5. 审计功能

DBA 对数据库的时候监督，叫审计

事后审计，事前评审

3.5.4 数据库的完整性

1.完整性约束条件：

对象是：关系，元组，列

如：主键不能为空，外键不可以修改等

2.完整性控制

分为实体完整性，参照完整性

如：非空，大于 0 等，本表的外键值必须对应外键表的主键而且该主键必须存在
外键级联操作，如级联删除等

4.触发器

trigger, 触发器就是在触发事件发生时必然执行结果事件

触发器有对实体完整性保护以及安全性保护

3.5.5 数据库性能

数据库性能优化

1.sql 优化

2.数据库参数调优

3.查询优化，如大表拆小表，建立索引，少用嵌套查询，联合表查询，
IN 语句中将出现频繁
的字段放在前面，尽量不用 OR 用 2 条语句来实现

3.6 数据库设计与建模

3.6.1 数据库设计阶段

1.规划

2.需求分析

3.概念设计，抽象需求中的对象

4.逻辑设计，建模如类图， powerdesigner 等

5.物理设计（建库），将模型创建到数据库中

3.6.2 设计约束和原则

设计约束包括如下：

查询响应时间的上限

系统破坏后的恢复时间

维护安全性和完整性必要的特殊数据

等等

3.6.3 实体联系模型

实体联系模型 E-R图

1.组成要素：实体（矩形），属性（圆形），联系（菱形）

2.实体联系类型：1对多，1对1，多对多（联系也是可以有独立的属性的）

3.E-R图集成

根据模块分别进行设计后，将多个 E-R图集成起来形成整体的 E-R图，可一次性集成也可逐步集成，会出现属性冲突（相同的属性不同的类型），命名冲突（表明相同等），结构冲突（属性与实体的变换，比如有的存的是一个属性，而有的存的是这个关联实体）等问题需要调整

4.E-R图转成关系模型

就是讲 E-R图中的实体转换成数据库表

3.7 常见的关系型数据库

oracle,mysql,db2(IBM),sqlserver(MS),sybase,informix(IBM)

3.8 分布式数据库系统、

分布式数据库特点：

安全性，多副本

可扩展性

降低成本

3.8.1 分布式数据库系统

1.分布式数据存储

1) 数据分配：

集中式，数据都放在同一地点

分割式，各个节点分别存储部分数据，所有节点的数据并集为全部数据

全复制式，各个节点存全部完整数据，也就是多个备份

混合模式，数据多块分别存储并有备份

2) 数据分片：

存储的数据不是完整的而是一个片段，多个片段组成完整视图

分片方式如下：

水平，元组分片

垂直，属性分片

混合，元组 + 属性分片

分片原则：

完整性

重构条件，随时重建

不相交条件，各个数据片段互相不重叠

2. 分布式数据库系统体系结构

全局外模式，用户视图

全局概念模式，逻辑模型

分片模式

分布模式，处理分散的分片，完整性处理

分布式数据库的透明性：

分片透明性

位置透明性

局部数据模型透明性

3.8.2 分布式数据库管理系统

功能：

接受用户请求

访问网络数据字典

进行分布式处理

通信接口功能

数据迁移

组成：

LDBMS(局部 DBMS)

GDBMS(全局 DBMS)

全局数据字典，提供全局公共属性

通信管理

总结：

局部管理，场地自治

全局管理，协调全局

3.9 数据仓库

3.9.1 联机分析处理

OLAP联机分析处理，是多维分析工具

分析方法：

- 1.钻取，层次维度加深
- 2.切片和切块
- 3.旋转，互换维度

联机分析处理实现方式

- 1.ROLAP, 关系型数据库为核心
- 2.MOLAP，多维型数据组织方式为核心
- 3.HOLAP;混合型

OLAP 就是多维度的方式组织数据，例如立方体，然后通过切片、切块、钻取、旋转的方式获取数组

3.9.2 数据仓库的概念

数据仓库特征：

- 1.面向主题，关系型数据库是面向事务处理及系统业务，而数据仓库是关心与某个主题相关的全部信息
- 2.集成的，业务系统的数据库是相互独立的或异构的，而数据仓库是将多个业务异构库，通过 ETL交换到同构的库中
- 3.相对稳定，关系型库经常更新，而数据仓库要保留一切数据定期进行更新与加载
- 4.随时间变化，关系型数据库只关心某时间段的数据，而数据仓库的数据时间跨度较大，几年或几十年，可反映历史变化的

数据仓库反映历史变化的表现

- 1.存储的数据久远
- 2.每个时间点的数据均存储，包括随时间发展的过程数据
- 3.包含时间元素如在某年发生的某时发生的或某秒发生的事件，例如，每年 5 月份大概率发生的事件是什么

数据仓库的分类

- 1.企业仓库，收集各个企业应用的全部操作数据
- 2.数据集市，对特定用户相关的有用的全部数据是企业仓库的子集
- 3.虚拟仓库，是数据仓库的视图集合

数据聚集与立方体

数据仓库是多维度的，也就是说是从多角度来观察某件事

3 个维度形成的数据块叫立方体，超过 3 个维度叫超立方或超维度

优点：提高 OLAP 的响应速度，减少重复计算

3.9.3 数据仓库的结构

1.参考框架

数据仓库的应用流程

业务系统 - (抽取 -数据准备区 -) ETL数据仓库 -数据集市 /只是挖掘库 -应用工具 -用户使用
全程由管理工具进行管理

2.数据仓库体系结构

数据源

数据 ETL-组织数据集市

OLAP，对集市及数据仓库进行联机分析处理

辅助管理工具集，包括查询工具，报表工具，分析工具数据挖掘工具等

3.10 数据挖掘技术

3.10.1 数据挖掘的 5 类功能

1.预测趋势和行为

2.关联分析，从一件事情联系到其他事情

3.聚类，把有共性的事件汇聚到一起

4.概念描述，对某类对象的内涵进行描述

5.偏差检测，检测与分析目标偏差较大的信息

3.10.2 数据挖掘常用技术

数据挖掘的方法

1.决策树

2.神经网络，参考人脑，采用权值分析

3.遗传算法，参考生物进化学，通过 "繁殖"下一代找出优质的一代

4.关联挖掘，就是从一件事联系到其他事

数据挖掘分析方法分类

- 1.关联分析，顺藤摸瓜
- 2.序列分析，事件的连续性
- 3.聚类分析，根据共同点进行分类
- 4.分类分析，根据分类的方式方法进行分析，给各个对象加分类标示
- 5.预测
- 6.事件序列分析，随时间推移，找出规律及趋势来预测未来走势

3.10.3 数据挖掘的流程

数据挖掘流程

- 1.问题定义
- 2.建立数据挖掘库
- 3.分析数据
- 4.调整数据
- 5.模型化
- 6.评价和解释，对建立的模型进行评审

3.10.4 数据挖掘技术的应用

- 1.空间数据挖掘
- 2.多媒体数据挖掘
- 3.文本数据挖掘

4 嵌入式系统

4.1 嵌入式系统概论

4.1.1 嵌入式系统的特点

- 1.系统专用性强
- 2.软硬件依赖性强
- 3.系统实时性强
- 4.处理器专用
- 5.多种技术紧密结合
- 6.系统透明性
- 7.系统资源受限

RTES实时嵌入式系统

概念特性：

- 1.逻辑正确性（功能正确性），结果正确
- 2.时间正确性，在规定的时间内完成
- 3.死线，最迟完成时间，如果超过这个时间会引起严重后果
- 4.实时系统，实时性

RTES系统分类

- 1.硬 RTES, 超过死线就失败，必须满足死线要求。
- 2.软 RTES, 超过死线不会失败，但是收益降低。

4.2 嵌入式的基本结构

4.2.1 硬件体系结构

嵌入式系统处理器分类

- 1.嵌入式微处理器，将系统软硬件集成到一块电路板上，又叫单板机
- 2.嵌入式微控制器，将系统集成到一块芯片中，又叫单片机
- 3.嵌入式数据自信号处理器，信号分析
- 4.嵌入式片上系统，系统功能模块及存储等集成到芯片上

4.3 嵌入式操作系统

4.3.1EOS特点与分类

- 1.微型化，小
- 2.代码质量高，精简
- 3.专业化，专业领域使用
- 4.实时性强
- 5.可裁剪可配置

实时性分类

- 1.实时嵌入式操作系 RTOS
- 2.非实时性嵌入式操作系统

4.3.3 多任务调度机制

- 1.任务，就是线程

- 2.任务对象，包括 ID,优先级，任务控制块，任务名称
- 3.多任务
- 4.调度器，用于调度任务执行
- 5.可调度实体
- 6.上下文切换，从一个任务切换到另一个任务，上下文切换耗时称作，上下文切换开销
- 7.可重入性，任务可以中断，以及继续执行
- 8.分发器，用于执行上下文切换工作

调度算法

- 1.优先级的抢占调度，静态与动态，静态优先级固定，动态优先级可变
- 2.时间轮转调度，抢占，可将运行中的时间片暂停后执行其他任务，执行完后再回复暂停的任务

4.3.4 内核对象

- 1.信号量
分类：二值信号量（并发），计数信号量（同步），互斥信号量（互斥）
- 2.消息队列
为了任务消息的同步，类似一个缓存队列
- 3.管道
用于任务同步，一个入口一个出口
- 4.事件
用于存储任务的寄存器，又叫事件寄存器
- 5.信号
事件发生时的软终端
- 6.条件变量
是事物申请资源等待的条件

4.3.5 内核服务

- 1.异常与中断
- 2.计时器
- 3.I/O 管理

4.4 嵌入式系统数据库

4.4.1EOS环境特点

- 1.设备随时移动
- 2.网络频繁断接
- 3.网络条件多样化
- 4.通信能力不对称

4.4.2EDBMS 关键技术

- 1.嵌入式 DBMS，移动端的独立数据库
- 2.同步服务器，用于同步嵌入式数据库与主数据库数据的同步
- 3.数据服务器，主数据库
- 4.连接网络，多种联网方式

移动 DBMS 关键技术

- 1.数据的一致性
- 2.高效的事务处理
- 3.数据的安全性

移动 DBMS 特性

- 1.微核结构，由于资源及空间有限
- 2.标准 SQL的支持
- 3.事务管理功能
- 4.数据同步机制
- 5.支持多种链接协议
- 6.完善的数据库管理功能
- 7.支持多种 EOS

4.5 嵌入式系统网络

- 1.FCS现场总线控制系统，运用现场总线链接各个移动系统成互通网络，可双向传输

4.6 嵌入式系统开发

- 1.开发环境：CPD(交叉平台开发方法)也就是开发在宿主机上，运行在嵌入式环境也可能是模

拟器

2.开发过程：项目计划，可行性分析，需求分析，概要设计，详细设计，程序建立，下载，调试，固话，测试运行

3.调试方法：直接调试（下载到嵌入式设备上调试），调试监控法（代码在宿主机上，运行在嵌入式设备上，通过宿主机与嵌入式设备连接后进行调试），在线仿真法（用 ICE 模拟目标机），片上调试法，模拟器法（在宿主机上安装模拟器软件）

4.7 嵌入式系统耗能问题

1.电源管理，动态电压调节

2.动态电源管理

5 数据通信与计算机网络

5.1 数据通信基础知识

5.1.1 信道特性

1.码元与比特

波特率是每秒传输多少个码元

比特率是每秒传输多少数据量

如果 1 波特传输 1 个码元，1 个码元携带 1bit 的数据则 波特 = 比特

2.信道极限信息传输速率

$C = W \log_2(1 + S/N)$ bps

S/N（带宽 / 噪声），当带宽远远大于噪声时为极限的传输速率，但是实际噪声的存在使得速度不会达到极限速率的

3.码元与调制技术

$N = 2^n$

1.ASK 2-1

2.FSK 2-1

3.PSK 2-1

4.DPSK 4-2

5.QPSK 4-2

码元种类数 N 与携带比特位数 n， $n = \log_2 N$

数据速率 $R = B \log_2 N$

B:码元速率 ,N：码元种类数

5.1.2 数据调制与编码

- 1.模拟通道传送模拟数据，方法有：调幅 AM(如收音机)，调频 FM，调相 PM
- 2.数字通道传送模拟数据，方法脉码调制 PCM，就是讲模拟信号量化后转化为二进制代码
- 3.模拟通道传送数字数据，如电话线拨号上网
- 4.数字通道传送数字数据

5.2 网络体系结构

5.2.1 网络分类

- 1.局域网 LAN，距离 100 米-2 万 5 千米，速度 4M-1G，成本低
- 2.广域网 WAN，距离远可跨城，速率低，成本高，高速广域网 ISDN综合业务数字网，ADSL非对称数字用户线路
- 3.城域网 MAN，距离介于局域网与广域网之间，主要技术 DQDB

5.2.2 网络互联模型 OSI/RM

- 1.物理层，传输 2 进制信号，CSMA/CD
- 2.数据链路层，IEEE802规范，传输帧信息
- 3.网络层，分组路由选择，IP,ICMP,IGMP,ARP,RARP
- 4.传输层，端到端链接，TCP,UDP
- 5.会话层，通信
- 6.表示层，数据表示，JPS,GIF,AVI
- 7.应用层，NFS,SNMP,DHCP,TFTP,FTP,HTTP,POP3,DNS

特殊的端口号

- 1.SMTP:25
- 2.POP3:110
- 3.DNS:53
- 4.WWW:80
- 5.FTP:21

256~1023 端口号由 UNIX 系统占用

5.2.4 网络地址与子网掩码

IP 地址结构

4 段 8 位二进制 01 组成，共 32 位

网络号与主机号

如 192.168.20.105

网络号 192.168.20.0

主机号 0.0.0.105

5 类地址 ABCDE

A 类：1~126 2 的 24 次幂 -2 个主机地址（2 个保留地址）首位 0 前 8 位网络号，后 24 位主机地址

B 类：128~191 2 的 16 次幂 -2 个主机地址（2 个保留地址）首位 10，前 16 位网络号，后 16 位主机地址

C 类：192~223 2 的 8 次幂 -2 个主机地址（2 个保留地址）首位 110，前 24 位网络号，后 8 位主机地址

D/E 类用的比较少，D 类最前面 4 位 1110，E 类最前面 5 位为 11110，部分网络号与主机号属于专门保留地址以及扩展地址

几个特殊地址

1. 主机号全为 1 的为广播地址如：192.168.10.255
2. 网络号全为 0，主机号表示本网地址如：0.0.10.125
3. 主机号全为 0，网络号就是本网的地址
4. 保留地址 255.255.255.255 0.0.0.0

子网掩码

A：255.0.0.0

B：255.255.0.0

C：255.255.255.0

例如：192.168.20.11/22

前 22 位为网络号则主机地址就是有 10 位，也就是有 2 的 10 次幂 -2 个地址

子网掩码为前 16 位均为 1，后 16 位中前 6 位为 1 因为网络号有 22 位 16+6=22

因此子网掩码为 11111111.11111111.11111100.000000=255.255.252.0

IPv6

IPv4 地址长度为 32 位，IPv6 地址长度 128 位

5.2.5 虚拟局域网

1. 虚拟局域网 VLAN 的功能，逻辑上划分网络使其物理上无需进行设置或增加设备

2. 划分方法：端口，MAC 地址，网络层协议，IP 组播组 VLAN，策略 VLAN，用户定义

5.3 通信设备

- 1.多路复用技术，就是把多个信号组合起来在一条物理信道上进行传输
- 2.分类 FDM 频分多路复用（按频谱划分），TDM 时分多路复用（按时间片划分）
- 3.模拟信号可结合 FDM 与 TDM 混合使用

5.3.1 传输介质

- 1.双绞线：距离短，速度快，价格便宜
分类：屏蔽与非屏蔽双绞线，屏蔽双绞线有一个金属屏蔽层抗干扰，减少辐射以及被监听
非屏蔽双绞线，无屏蔽外套，轻易安装，可弯曲，灵活性高，适用于综合布线
按照口径由小到大：5 类最细速率为 100M，超 5 类 1000M，6 类最粗，2000M（2 倍超 5 类速率）
- 2.同轴电缆：距离远，速率 50M 左右，抗干扰能力强，价格介于双绞线与光纤之间
分类：粗同轴电缆，细同轴电缆
- 3.光纤：速度快，不受电磁干扰（因为是光信号），价格贵，距离远，衰减小
分类：多模 200M-3G，单模 3G-50G，多模比单模成本低

5.3.2 网络设备

- 1.网卡：用于收发信号，工作在数据链路层
- 2.集线器：工作在数据链路层，用于连接局域网，平分带宽换式，共享带宽
- 3.重发器，中继器：用于放大信号
- 4.网桥：工作在数据链路层，网桥用于连接 2 个不同网段，可进行数据过滤
- 5.交换机：用于连接网络，连接数据链路层和物理层协议转换
分类：2 层交换机工作在数据链路层，起到多端口网桥的作用，三层交换机工作在网络层，利用 IP 地址进行交换，相当于带路由功能的二层交换机
- 6.路由器：工作于网络层，用于路径选择
- 7.网关：又叫协议转换器，用于 2 个不兼容的系统在高层进行协议转换
- 8.调制解调器：又叫猫，是进行模拟信号与数字信号互相转换

5.4 网络接入技术

- 1.ATM
- 2.帧中继
- 3.ISDN
- 4.同步光纤网络
- 5.DDN

- 6.FTTX光纤到 X
- 7.电话线拨号接入
- 8.HFC同轴和光纤接入
- 9.无线接入
- 移动 WCDMA 联通 CDMA2000 电信 TD-SCDMA

6 多媒体

6.1 多媒体基础

6.1.1 多媒体计算机

- 1.音频卡
- 2.视频卡
- 3.采集卡
- 4.扫描仪
- 5.光驱

6.1.2 媒体的分类

- 1.感觉媒体，音频，视频等
- 2.表示媒体，文本，图像
- 3.表现媒体，鼠标键盘等输入输出设备
- 4.存储媒体，硬盘光盘， CD-ROM,DVD-ROM
- 5.传输媒体，电缆光缆等

6.1.4 多媒体集成语言 SMIL

6.2 压缩编码技术

- 1.空间冗余，类似的码
- 2.时间冗余，如视频的相同帧
- 3.知觉冗余，降低质量，减少帧数，在人们感官不可察觉的程度
- 4.信息熵冗余，数据携带的信息量
- 5.结构冗余，如布纹图，草席图等更改结构来达到压缩
- 6.知识冗余，将规律性的结构压缩

压缩类型

- 1.无损压缩，压缩解压可逆，不是真，但是压缩比例小 2:1 5:1
- 2.有损压缩：压缩了熵，减少信息量，可能会失真，但是压缩比例大声音 4:1 8:1 视频 100:1 400:1

压缩域

- 1.时间域，传输时间短
- 2.频率域，可并行多传
- 3.空间域，压缩体积减少存储
- 4.能量域，降低发送功率

6.2.2 数据压缩标准

- 1.JPEG(联合图像专家组),采用 DCT(离散余弦变换)压缩方式，用于图片压缩
- 2.MPEG(动态图像专家组)，压缩比高可达 200：1，用于音视频压缩，标准（ MPEG-1 普通音视频， -2 高清， -3 用于音频， -4 视频电话， -7 在线播放， -21 兼容不同协议）
- 3.DVI(数字视频接口)，类似 MPEG-1
- 4.H.26,类似 MPEG，用于视频会议等
- 5.H263，用于低带宽条件下

6.3 音频数据

6.3.1 音频

关键要素：音强，音调，音色

6.3.2 音频的存储与传输

存储步骤：

- 1.采样，采集模拟声音的波形上采集幅度值，采样的频率为最大频率的 2 倍
- 2.量化，将电压用数字进行量化，单位一般为 bit
- 3.编码，将声音数据编码成计算机格式

每秒存储字节数的计算公式

文件的字节数 bit= 采样频率（ Hz） * 量化采样位数 * 声道数 /8（如果单位为 Bit 则不用除 8）

6.3.3 音频格式

- 1.WAVE .wav
- 2.MOD .mod
- 3.Layer-3 .mp3
- 4.Real Audio .ra
- 5.CD Audio .cda
- 6.MIDI .mid
- 7.CMF .cmf

6.4 色彩空间

6.4.2 颜色空间

- 1.三原色，红绿蓝（ RGB）

6.4.3 图形与图像

关键指标：分辨率（像素点数），点距（像素点之间的距离），深度（色彩数或灰度，如 n 位色说明是有 2 的 n 次幂种颜色， 26 位色就是 2 的 26 次幂种色彩）

- 1.图像，称作位图或点阵图，格式有 BMP，TIF 等
- 2.图形，又叫矢量图，石油直线，圆，圆弧等任意曲线和图标绘制而成，格式有 3DF，等

1 个字节 =8 位

$256 \text{ 色} = \log_2 256 = 2$ 的 8 次幂也就是 8 位，如果分辨率是 640×480 那么这个图片的大小就为 $640 \times 480 \times 8 / 8 = 307200 \text{B} = 300 \text{K}$

3.图形文件格式

- 1.BMP
- 2.DIB
- 3.GIF
- 4.DIF
- 5.JPEG

视频文件格式

- 1.Quicktime 苹果视频
- 2.AVI
- 3.WMV

6.5 流媒体

传输方式

- 1.顺序流传输，不可快进只能按顺序加载，加载完的可以回放及快进
- 2.实时流传输，穿一帧看一帧，可快进，因为有时间刻度，可根据快进到的时间点开始传输

流媒体协议

- 1.PNA,Real 专用流媒体协议，采用 UDP 协议
- 2.MMS，微软的
- 3.RTP, 实时传输协议
- 4.RTCP, 实时传输控制协议
- 5.RTSP, 实时流协议

7 系统配置与性能

7.1 系统性能计算

7.1.1 响应时间，

发出请求到返回响应的时间间隔

影响响应时间的可能因素：

- 1.业务相关
- 2.业务组合
- 3.用户数量

响应时间点

- 0.1 秒最佳
- 1 秒正常
- 10 秒最大忍受

7.1.2 吞吐量

是系统单位时间内处理任务能力，也就是说在单位时间内接受处理任务数量

性能计算公式

- 1.平均利用率 $p = \text{平均到达事务数} / \text{平均处理事务数}$
- 2.平均响应时间 $= \text{平均处理时间} / (1 - p)$

7.1.3 系统可靠性

1.平均无故障时间

可靠度 $R = \text{无故障时间} / \text{总时间}$

失效率 $= \text{失效时间} / (\text{机器数} * \text{运行总时间})$

平均无故障时间 $MTTF = 1 / \text{失效率}$

2.平均故障修复时间 $MTTR$

3.平均故障间隔时间 $MTBF = MTTR + MTTF$; $MTBF \gg MTTR$ 犹豫 $MTTR$ 一般很小，所以故障间隔时间就约等于平均无故障时间

4.系统可用性

可用性 $= MTTF / (MTTF + MTTR) * 100\%$

串联系统

输入 R_1, R_2, \dots, R_n

可靠性 $R = R_1 * R_2 * R_3 * R_4 * \dots * R_n$

失效率 $= 1 - R = 1 - R_1 * R_2 * R_3 * R_4 * \dots * R_n$

并联系统

$R = 1 - (1 - R_1) * (1 - R_2) * \dots * (1 - R_n)$

7.2 系统性能设计

7.2.1 性能调整过程

收集、分析、配置、测试 4 个循环过程

性能调整准备工作

1.识别约束

2.制定负载

3.设置性能指标

7.2.2

加速比 $= \text{不使用增强部件完成时间} / \text{使用增强部件完成时间}$

7.2.3 负载均衡

NAT网络地址转换采用了负载均衡技术，就是多个内网地址转换为一个外网地址实现多个内网地址通过一个外网地址进行上网

7.3 系统性能评估

1.时间频率法

概念：

时钟频率（主频）：反映了计算机的工作效率，主频越高计算机工作速度越快

时钟周期：时钟频率的倒数，一个时钟周期中 CPU只完成一个操作

机器周期：由多个时钟周期组成，如取指令、存储器读、存储器写，每一个基本操作称作一个机器周期，一个机器周期由多个时钟周期组成

指令周期：执行一条指令所需要的时间，一个指令周期由多个机器周期组成

例子：

一台微机主频为 20MHz，如果 2 个时钟周期 =1 个机器周期，3 个机器周期完成一条指令则：

时钟周期 =1/主频 =1/20M=50ns；机器周期 =2*50ns=100ns；指令周期 =3*100ns=300ns；指令平

均执行速度 =1/300ns=3.3MIPS（每秒百万条指令数）

1s=1000ms=1000000 μs=1000000000ns

1 秒=1 千毫秒 =100 万微秒 =10 亿纳秒

2.指令执行速度法

MIPS 每秒百万指令数

MFLOPS每秒百万浮点操作次数

3.等效指令速度法

不是很准确

4.数据处理速率法

PDR(已淘汰)：对 CPU和内存处理数据的速度的评估

5.CTP(综合理论性能法)

6.基准程序法

根据多种基准程序进行测试，每个程序侧重点不同，是目前效果比较好的性能测试方法

基准程序包括：

Khrystone

Linpac

Whetstone

SPEC

TPC

7.4 系统故障模型

系统故障的表现形式：

1.永久性

2.间歇性

3.瞬时性

系统故障级别模型

- 1.逻辑级
- 2.数据结构级
- 3.软件故障和软件差错
- 4.系统级

级别越高危害越大，代价越高

7.5 系统可靠性模型

- 1.时间模型
- 2.故障植入模型
- 3.数据模型

7.6 冗余技术

1.结构冗余

静态冗余：错误可被其他模块的正确结果覆盖

动态冗余：系统每个模块有读个热备模块，动过系统动态监测与重构对故障模块进行替换

混合冗余：兼静态动态冗余的长处

2.信息冗余

增加纠错的信息如纠错码

3.时间冗余

重复执行指令

4.冗余附加技术

故障恢复策略

- 1.恢复到后状态，就是越过故障点继续执行，故障通过日志记录
- 2.恢复到前状态，就是恢复到故障前一个正确状态。

7.6.2 软件容错

- 1.恢复块方法，主块验证测试 -不合格需要找后备块 -检测 -如果还不通过继续找直到合格后输出正确结果
- 2.N 版本程序设计，采用多个版本的使用不同方法实现相同功能的程序进行计算通过表决器得出正确结果
- 3.防卫式程序设计，错误发生时可以及时撤销错误状态，恢复到一个已知的正确状态。

7.6.3 集群技术

1.集群分类

高性能计算科学集群，并行计算

负载均衡集群，降低负载压力

高可用性集群，故障解决与预防能力强，可用性强，不易宕机

2.集群的硬件配置

1.镜像服务器双击，热备

2.双击与磁盘阵列，存储高可用

3.光纤通道双击双控集群系统

8.信息系统基础

8.1 信息特征

1.客观性

2.普遍性

3.无限性

4.动态性

5.依附性

6.变换性

7.传递性

8.层次性

9.系统性

10.转化性

8.2 信息与信息系统

8.2.1 系统的特性

1.整体性：由多个元素有机结合的整体

2.层次性：如系统与子系统

3.目的性：系统功能，需要实现的功能

4.稳定性：持续工作

5.突变性：状态变换

6.自组织性：

7.相似性：同构异构系统

8.相关性：元素间相关

9.环境适应性

8.2.2 系统理论

1.系统的整体性原理

2.系统的整体突变原理：非加和原理

3.系统的层次性原理

4.系统的开放性原理

5.系统的目的性原理

6.系统环境互塑共生原理：功能与压力

7.系统的秩序原理：有序的系统

8.系统的生命周期原理：系统生命周期

8.2.3 系统工程

霍尔三维结构

1.逻辑维步骤：步骤无绝对顺序 ,各个步骤可能要反复多次

明确问题

系统指标设计：设计评定指标

系统方案综合：将多个方案综合评论

系统分析

系统选择：从多个方案中选择最优

决策：决定使用哪个方案

实施计划

2.时间维阶段

规划阶段

拟定方案

研制阶段

生产阶段

安装阶段

运行阶段

更新阶段

3.知识维

完成逻辑维与时间维各个步骤需要的专业知识，把业务知识进行分类如行业等

8.2.4 信息系统工程

1. 信息系统的基本

数据的采集和输入

数据的传输：信息源（发送者） - 编码 - 噪声 - 译码 - 信宿（接收者）

信息的存储

信息的加工：信息的处理如排序，归并等等

信息的维护：增删改查

信息的使用

2. 信息系统分类

数据环境分类

应用层次分类： 战略级（企业高层管理）、战术级（企业中层管理）、操作级（企业业务部门）、事务级（业务管理人员）

3. 信息系统建设

信息系统生命周期：

系统规划：输出系统设计任务书，步骤：开发请求 - 初步调查 - 可研 - 审批

系统分析：根据系统设计任务书内容进行分析， 系统分析主要目的是回答系统做什么的问题；

步骤：详细调查 - 逻辑设计 - 评审

系统设计：回答怎么做的问题，步骤：总体设计 - 详细调查 - 评审

系统实施：编码测试 - 系统转换 - 验收

系统运维：系统维护 - 系统评价

各个开发阶段人力需求：

随着开发阶段的进行，系统开发人员的参与度降低，用户参与度加强

4. 信息系统的发展

诺兰提出信息系统发展阶段分为：

初装

蔓延

控制

集成

数据管理

成熟

以上阶段由上到下预算增加

8.3 信息系统建设

8.3.1 企业规划方法 **BSP**原则

- 1.信息系统必须支持企业的战略目标
- 2.信息系统的战略应当表达出企业的各个管理层次的需求
- 3.信息系统应该向整个企业提供一致的信息
- 4.信息系统应该适应组织机构和管理体制的改变
- 5.信息系统战略规划由总体信息系统的子系统实现

BSP对于系统规划是自上而下，对于实现是自下而上

BSP方法的目标

- 1.弱化部门的狭隘利益
- 2.基于业务活动过程建立不受机构变化而失效
- 3.提高效率，提供数据处理管理
- 4.增加信息
- 5.改善部门与用户间的关系

BSP方法的研究步骤

- 1.研究项目的确定
- 2.研究的准备工作
- 3.研究开始阶段
- 4.定义企业过程
- 5.定义数据类
- 6.分析现存系统支持
- 7.确定管理部门对系统的要求
- 8.提出判断和结论
- 9 评价信息资源管理工作
 - 定义信息总体结构
 - 确定总体内的优先顺序
- 10.制定建议书和开发计划
- 11.研究成果报告

信息资源管理基本内容

- 1.资源管理的方向和控制
- 2.建立企业信息资源指导委员会
- 3.建立信息资源的组织机构

BSP研究可以得出的结果

- 1.信息结构
- 2.信息系统管理
- 3.分布信息系统规划
- 4.总体结构优先顺序

8.3.2 战略数据规划方法

詹姆斯马丁的观点

1.自顶向下的规划

步骤

企业模型的建立

确定研究的边界

建立业务活动过程

实体和活动的确定

对所得规划结果进行审查

建议全局规划时间 6 个月内

2.企业模型的建立

3.主题数据库

特征：

面向业务主题 ,而不是面向业务

信息共享，所有部门共享

一次一处输入系统，而不是多次多出输入

由基本表组成（满足第三范式 3NF）

四类数据环境：文件环境，应用数据库环境，主题数据库环境，信息检索系统环境

4.战略数据规划的执行过程

企业的实体分析，有哪些实体也就是表对象

实体活动分析，

企业的重组

亲合度分析，实体间的关联度或依赖度

分布数据规划

8.3.3 信息工程方法（规划与实施的方法）

1.信息工程步骤

信息战略规划

业务领域分析

业务系统设计

技术系统设计

系统构成

系统转换

系统运行

2.信息战略规划

评估企业的信息需求

建立企业总体信息结构

建立企业业务系统结构

建立企业技术结构

提交信息战略规划

信息战略规划任务：

制定信息战略规划项目的计划

初始评估

定义信息结构

评估当前的环境

确定业务系统结构

完成信息战略规划项目，提交规划报告

3.数据和资料的收集

收集的资料：

有关制定企业计划的资料

有关组织结构的资料

有关业务活动的资料

现有系统的环境资料

当前技术环境的资料

4.信息战略规划报告

报告内容：

摘要内容，不超过 5 页

信息战略规划所涉及的范围

企业的业务目标和战略重点

信息技术对企业业务的影响

对现有信息环境的评价

推荐的系统战略

推荐的技术战略

推荐的组织战略

推荐的行动计划

报告主要内容， 40-70 页

阐述总体内容

业务环境描述

评价现有信息环境

通过可选择方案和推荐的信息结构

最后给出推荐的行动计划

8.4 信息系统规划方法

8.4.1 第一个阶段

第一阶段：数据处理为核心，围绕职能部门需求进行规划

包括 3 个规划方法分别如下：

- 1.关键成功因素法 CSF根据分析企业成功关键因素，围绕这些因素进行分析规划；抓住主要矛盾突出重点
- 2.战略集合转换法 SST: 根据各种人员的要求进行分析并转化为系统的目标
- 3.企业规划方法 BSP: 强调目标

以上三种方法的综合分析：可结合使用，先用 CSF确定企业目标，再用 SST完善目标，最后用 BSP校核两个目标并确定系统结构

8.4.2 第二阶段

第二阶段：以企业内部管理信息系统为核心，围绕企业整体需求进行规划

规划方法：

- 1.战略栅格法
4 个格子分别为支持型，工厂型，转变型，战略型
- 2.战略系统规划法 SSP

8.4.3 第三阶段

第三阶段：集成为核心，围绕企业战略需求进行规划

规划方法：

- 1.价值链分析法 VCA,步骤如下：
分解价值链为作业、成本、收入和资产并分配到作业中
确定引起价值变动的作业
分析价值链中各个节点的关系
利用分析结果重组或改进价值链

VCA的作用：作为一种审计工具

- 2.战略一致性模型 SAM

概念：就是把企业运营与 IT 的战略达成一致性

方法：评价信息系统与竞争战略之间的对应程度，识别信息系统对企业战略的潜在影响，提出替代途径，重新制定信息系统战略

步骤：识别竞争环境，判断选择理论模型，提出解决途径及方案，根据模型进行方案选择

8.5 信息化基础

6 要素：上应（信息技术应用），下技（信息产业技术），左人（信息化人才），右规（信息化法规政策），资源核心（信息资源），网络基础（信息网络）

8.5.1 企业资源计划 ERP

来源于 MRP-II（制造资源计划 II）

1.ERP的功能

财会管理

生产控制管理

物流管理

人力资源管理

2.ERP的实施

三个时期：前期准备基础数据和标准化，中期业务重组，后期实施适应期

实施过程：

建立团队

明确目标制定实施计划

相关人员培训

需求分析

需求建模，原型分析

系统实施

运行试验

制定技术解决方案

调试培训测试

上线准备

运行

系统优化

8.5.2 客户关系管理 CRM

8.5.3 供应链管理 SCM

供货的整个渠道相关内容管理

8.5.4 产品数据管理 PDM

用来管理产品的全部信息如配置、文档、设计文件、结构等

8.5.5 产品生命周期管理 PLM

8.5.6 知识管理

分类显性知识（可文字描述） ，隐性知识（个人技术等）

8.5.7 商业智能 BI

包括技术：

数据仓库：面向主题，同构的；结构包括数据源 -ETL数据仓库；数据集市

OLAP联机处理分析

数据挖掘：钻取，数据立方体，切片，切块；遗传学（最优后代） ，神经网络（加权）
分析预测

8.5.8 企业门户

1.信息门户 EIP: 信息发布，是信息集成

2.知识门户 EKP

3.应用门户 EAP: 应用集成

8.5.9 电子政务

概念：通过信息技术优化政府机构业务流程， 需要进行业务重组， 采用不同的方式及流程实现政府工作

G2G 政府对政府

G2E 政府对公务员

G2B 政府对企业

G2C 政府对公民

8.5.10 电子商务

B2B 企业对企业

B2C企业对个人

C2C个人对个人

8.6 决策支持系统 DSS

分类：

结构化决策（有完整的决策模型）

非结构化决策（十分复杂无法用模型来决策）

半结构化决策

步骤：

发现问题形成决策目标

用概率定量描述各种方案结局可能性

决策人员定量估计、评价、评估

综合分析，最终决定方案取舍

决策一般是以迭代的方式完成，而很难一次成功

DSS的功能归纳如下：

- 1.整理与提供与本决策问题相关的数据
- 2.收集存储相关数据
- 3.收集提供有关项的反馈信息
- 4.能够存储于决策有关的模型
- 5.存储各种数学与运筹学方法
- 6.能灵活运用模型
- 7.提供人机交互图形化入口
- 8.加工速度与响应时间在可容忍接受范围
- 9.支持但不能替代上层决策

8.6.2DSS基本结构

数据库部分，是 DSS的数据基础部件，对基础数据的存储、管理、提供用户决策。

模型部分，是 DSS最复杂最难实现的部件，它是提供用户决策模型，因为 DSS是由模型驱动的

推理部分，是存储推理方法并使用，是由知识库管理系统，推理机系统组成

人机交互部分，是人机交互的界面，为使用者提供服务

SDSS综合决策支持系统组成：

数据仓库

联机分析处理

数据挖掘

模型库

数据库
知识库

8.6.3DSS与 MIS 的比较

DSS:

高效，质量，决策，适应性强，主观性强，模型驱动，解决非结构化问题

MIS

高效，快速，信息处理，稳定协调，客观性强，数据驱动，解决结构化问题

8.7 业务流程重组（ BPR ）

BPR不是对业务流程进行简单的调整，而是重新规划重新设计，脱胎换骨的过程

1.指导原则

组织机构设计要围绕企业产出而不是任务

使用过程输出的人来执行过程

信息处理工作整合到信息生产过程中去

分散资源集中处理

平行活动紧密连接

在基层活动中创建决策点

保持信息一致，在信息源头获取信息

2.实施建议

BPR项目启动时创建领导机制

企业人员参与 BPR工作，最好全员参与

调研范围要广泛全面

进行成本效益分析

避免无法衡量的部分

加强交流

重组全部流程

3.实施框架

KBSI公司的 BPR框架

多层的 BPR实施框架，分为观念重建层，流程重建层，组织重建层

8.7.2BPR的实施步骤

1.项目启动

2.拟定计划

- 3.团队组建
- 4.分析目标流程
- 5.重新设计目标流程
- 6.实施
- 7.持续改进
- 8.重新开始

8.7.3BPR与信息系统规划

基于业务流程的信息系统规划主要步骤：

- 1.战略规划，明确企业的战略目标，信息系统目标与战略目标一致
- 2.流程规划，规划企业业务流程
- 3.数据规划，规划流程产生的数据
- 4.功能规划，识别功能模块
- 5.系统实施

8.8 企业应用集成 EAI

8.8.1 企业内的集成

- 1.界面集成：
- 2.平台集成
- 3.数据集成：通过数据交换数据中心等方式
- 4.应用集成
- 5.过程集成

8.8.2 企业间的集成

如使用 webservice 等通用服务方式

8.8.3 集成模式

- 1.面向数据信息的集成，采用适配器技术
- 2.面向过程的集成技术，提供过程接口
- 3.面向服务 SOA,WEBSERVIC技术，用于异构系统集成的解决方案

9 软件工程

解决软件危机，软件危机就是软件成功率低，开发周期比发展周期慢，跟不上发展节奏

9.2 软件开发生命周期

主要过程：

- 1.获取过程
- 2.供应过程
- 3.开发过程
- 4.运行过程
- 5.维护过程

支持过程：

- 1.文档编制过程
- 2.配置管理过程
- 3.质量保证过程
- 4.验证过程
- 5.确认过程
- 6.联合评审过程
- 7.审计过程
- 8.问题解决过程

组织过程：

- 1.管理过程
- 2.基础设施过程
- 3.改进过程
- 4.培训过程

9.3 软件开发方法

9.3.1 净室方法（形式化方法）

净室软件工程（也叫净室方法）是软件开发的一种形式化方法，它可以生成高质量的软件，是一种严格的软件工程方法

净室方法的三种盒类型：

- 1.黑盒
- 2.状态盒
- 3.清晰盒

9.3.2 结构化方法（非形式化）

SA 结构化分析

SD 结构化设计

SP 结构化程序设计

1. 结构化方法的基本原则

- 1) 面上用户的观点，时刻与用户进行沟通获取用户需求
- 2) 严格区分工作阶段，每个阶段有明确的工作与成果
- 3) 按照系统的观点，自顶向下完成工作
- 4) 充分考虑到可变化性
- 5) 工作成果文档化

2. 结构化分析 SA

步骤：

- 1) 分析当前的情况，绘制物理模型 DFD 数据流图
- 2) 推导出等价的逻辑模型的 DFD
- 3) 设计心得逻辑系统生成数据字典和基元描述
- 4) 建立人机接口，可供选择的物理模型 DFD
- 5) 确定各种方案的成本风险，并进行分析
- 6) 选择方案
- 7) 建立完整的需求规约 SRS

3. 结构化设计 SD

步骤：

- 1) 评审细化数据流图
- 2) 确定数据流图类型
- 3) 根据数据流图设计模块结构的上层
- 4) 基于数据流图分解高层模块，设计中下层模块
- 5) 优化模块结构，得到更为合理的软件结构
- 6) 描述模块接口

SD 方法的设计原则：

- 1) 一个模块只完成一个功能（高内聚）
- 2) 使用过程语句调用其他模块（通过接口方式）
- 3) 模块间传送参数作为数据使用
- 4) 模块间共用的信息尽量少，也就是要低耦合

4. 结构化方法的缺点

开发周期长

重视系统功能，弱数据结构

对于非结构化系统，很难定位其功能

9.3.3 面向对象方法

主要工具 UML

1.Coad/Yourdon 方法

组成：

OOA 面向对象的分析，包括 5 层次，5 步骤

OOD 面向对象的设计，贯穿 OOA 的 5 层次 5 步骤

2.Booch 方法

后被合并为 UML

3.OMT 方法

后被合并为 UML

4.OOSE

在 OMT 的基础上进行补充

后被合并为 UML

Booch,OMT,OOSE统称为 UML

9.3.4 原型法

结构化方法和面向对象方法的共同点：在系统开发初期有明确的需求功能以及系统边界

因此原型法诞生，在系统开发初期开发出系统原型可在前期对系统有直观的认识

可以通过测试原型获取到更多的不足之处，提高后期软件的质量

1.原型的分类

水平原型（行为原型）：只有功能菜单和静态页面，没有实现功能，用于细化需求

垂直原型（结构化原型）：实现的部分功能，用于实现复杂算法

抛弃型原型（探索型原型）：用于进行需求分析以及需求理解，达到目标后，原型就没有利用价值而被抛弃

演化型原型：会时刻更新，是螺旋模型的一部分，该模型必须易于升级与优化，适用于 web 项目

2.原型类型的选择

1) 系统结构：联机事务处理系统适合使用原型法，批处理、批量修改等结构不适用

2) 逻辑结构：有结构的功能型系统适合，而大量算法的系统不适合

3) 用户特征：前期需求不确定不明确，用户愿意投入原型成本的用户适合使用

4) 应用约束：对于已有系统扩展功能不适合使用

5) 项目管理：项目负责人愿意使用

6) 项目环境：根据实际环境来选择是否适用

对于功能复杂，前期用户需求没有整体概念的大型系统，适合使用原型法

3.原型生命期

步骤：

- 1) 快速分析
 - 2) 构造原型
 - 3) 运行评价
 - 4) 修正改进
 - 5) 判定是否完成
 - 6) 判断是否需要详细说明
 - 7) 细部说明，对于不能用原型表现的要通过描述来讲清楚
 - 8) 判定原型效果
 - 9) 整理原型，提供文档
- 原型就是动态可视化需求规格说明

4.原型开发技术

总结：原型法适合于前期用户需求不明确的项目，在开发原型过程中用户全程参与并及时提出建议大大提高了最终系统的客户满意度

9.3.5 逆向工程

1.再工程

步骤：

逆向工程，通过已有软件系统的源码，文档等分析其设计理念，从而挖掘出其优点
软件重构，利用上一步分析的代码优点在原有代码的基础上重新构建，提高质量
正向工程，根据上两步的输出进行正常的软件功能方法进行开发

9.4 软件开发模型

9.4.1 瀑布模型

也叫生命周期法，是结构化方法，顺序执行，一次流程通过后得出最终产品
阶段如下：

- 1.软件计划
- 2.需求分析
- 3.软件设计
- 4.程序编码
- 5.软件测试
- 6.软件维护

优点：利于大型软件开发中人员的组织与管理，提高大型软件开发的效率和质量

缺点：必须前期需求明确，产品成型时间长，只有在最后才能看到产品的全景

由于是线性执行，过程中每个阶段出现严重问题都会成为瓶颈而是整个项目失败

前期完全确定用户需求是比较困难甚至是不可能的

瀑布模型适用于前期需求明确，很少变更的项目

9.4.2 其他经典模型

1.演化模型（快速原型法或变换模型）：在初期快速开发出原型，尽快让原型与用户接触，前期使用户可以提出有价值的建议

2.螺旋模型：适用于高风险的项目，他与瀑布模型相结合，一个螺旋模型多个阶段，每个阶段都需要风险评估

3.喷泉模型：开发、分析、设计没有明确的阶段可迭代执行

4.智能模型：利用专家系统提供建议

5.增量模型：多级瀑布模型重复执行，不断完善，每个阶段输出实现部分功能的可发布的版本，下一个版本基于上一个版本增量发布

6.迭代模型：类似增量模型，只不过迭代模型是不断完善的过程，而增量模型是增量发布的是功能模块，比如先开发 3 个功能模块即可发布，后期在开发 2 个模块增量发布后就有 5 个模块可用了

7.构建组装模型（CBSD）：基于构件的开发，利用模块化方式进行迭代开发

9.4.3V 模型

V 模型是指在开发各个阶段中融入测试，而不像瀑布模型只有在后期才介入测试。

开发阶段与测试阶段对应关系：

需求分析：验收测试

概要设计：系统测试

详细设计：集成测试

编码：单元测试

9.4.4 快速应用开发（RAD）

基于构件开发，利用构件复用的特性达到快速开发

缺点：必须全部功能模块都可以模块化构件化

必须可以快速的确定需求并与客户达成一致

不适合技术风险高的项目

9.4.5 敏捷方法

开发者角度的关键点：

短平快会议

小版本发布

较少的文档

合作为重

客户直接参与

自动化测试

适应性计划调整

结对编程

管理角度的关键点：

测试驱动开发

持续集成

重构

极限编程 XP：是一种轻量敏捷、高效、低风险、可预测的适用于小型中型软件开发团队，客户需求模糊多变

总结：敏捷方法适用于小规模软件的开发团队，对于团队的要求比较高，在实际工作中可以借鉴其思想

9.4.6 统一过程

UP 是 Rational 公司提出，IBM 收购 Rational，因此也叫作 IBM RUP
使用 UML，基于构件的、用例驱动、以体系结构为中心、迭代增量

4 个阶段：

初始、细化、构建、交付

9.5 系统规划与问题定义

总体规划的任务：

制定软件发展战略

制定组织的主要信息需求，形成总体结构方案，安排项目开发计划

制定系统建设的资源分配计划

阶段：

初步调查当前系统

分析确定系统目标
分析子系统组成及功能
拟定系统实施方案
可行性分析
编写可行性分析报告

9.5.1 可行性分析

1.可行性分析内容

技术可行性
经济可行性
操作可行性
社会可行性 /法律

2 可行性分析步骤（系统分析师的职责）

- 1) 核实问题定义与目标
- 2) 研究分析现有系统
- 3) 为新系统建模

建模技术：系统上下文关系范文图，系统与外界实体的关系图

实体关系图 E-R图

用例模型

域模型，实体类的描述

IPO 输入 /处理 /输出

- 4) 用户复核，建模之后需要用户复核正确性
- 5) 提出并评价解决方案，过滤掉技术上无法实现，经济上无法满足，法律上不符合规定的方案
- 6) 确定最终推荐的解决方案
- 7) 起草开发计划，初略的开发计划
- 8) 提交可行性分析报告，参考国家标准 GB/T8567-88 的格式内容

可研报告大致内容：

引言

可行性研究的前提

对现有系统的分析

建议的系统方案

可选的其他系统方案

投资及效益分析

社会因素方面的可行性

结论

9.5.2 成本效益分析

1.货币的时间价值 F (n 年后的价值) P (现在的价值) i (利率) n (年份) , 如果贷款需要每年支付利息

利滚利 $F=P*(1+i)^n$?

折现值 / 贴现值 $P=F/(1+i)^n$?

折现系数 $=1/(1+i)^n$?

2.净现值分析法

$NPV=n$ 年总收益的贴现值 - n 年投资的总贴现值

NPV 越大越好

NPV 小于 0 证明这个投资方案会亏损, 不可取

3.现值指数法

$NPVI=n$ 年总收益的贴现值 / n 年投资的总贴现值

$NPVI$ 越大越好

$NPVI$ 小于 1 证明这个投资方案会亏损, 不可取

4.内含报酬率 IRR

$IRR=i_1+(i_2-i_1)*|NPV_{i1}|/(|NPV_{i1}|+|NPV_{i2}|)$

IRR 是极限的贴现率, 超过这个利率就会亏损

5.案例

投资年份与收益年份编号, 从 0 开始编数, 根据念书进行计算贴现率, 通过贴现率计算每年的贴现值, 然后将历年投资的贴现值总和与历年收益的净现值总和进行计算

6.投资回收期

动态与静态, 动态用贴现值计算, 静态用现值计算

回收期 = 盈利年份 - 1 + | 盈利年前一年盈利和 - 总投资成本 | / 盈利年当年利润

总结: 就是到那一年能收回成本

7.投资回收率

投资回收率 = 1 / 投资回收期

8.盈亏平衡点

卖多少本能收回成本

盈亏平衡点 $BEP=固定成本 / (总销售额 - 总可变成本)$

9.5.3 系统建模

概念:

物理模型：就是实际的系统或人工处理流程，由于是实际的系统因此各项细节都有

逻辑模型：是设计模型如 UML 的设计模型，逻辑模型只说明系统的功能及处理的数据，而忽略细节

建模过程：

- 1.获得当前系统的物理模型，了解当前系统“ 怎样做事情 ”
- 2.抽象出当前系统的逻辑模型 ,分析出当前系统“ 是做什么的 ”
- 3.建立目标系统的逻辑模型，根据当前系统的逻辑模型导出目标系统的逻辑模型，也就是目标系统“ 做什么 ”
- 4.建立目标系统的物理模型，根据逻辑模型构建物理模型

9.5.4 问题定义

步骤：

- 1.在问题定义上达成共识：将问题标准化的格式写出来，让项目干系人、开发、客户的需求问题达成一致。
- 2.理解问题的本质：理解问题的根本原因，使用因果图、鱼骨图、帕累托图等工具进行分析
- 3.确定项目干系人：确定与项目相关的全部角色，了解干系人的需要以及对问题的不同看法
- 4.定义系统的边界：系统与外界的边界，利用用例图等方法
- 5.确定系统实现的约束：考虑到各种对系统的限制如预算、人员、收益、操作、性能、社会等一切约束条件

问题定义的 3 个方面

- 1.目标 ,要达到的目的
- 2.功能，用什么功能来实现目标
- 3.非功能需求，易用性、性能需求等

9.6 需求工程

1.需求开发

步骤：

需求获取：调查问卷，小组会议，文档考古，现场观摩等

需求分析：业务需求分析成开发需求

需求定义：编写需求规格说明书

需求验证：与客户确认签字

2.需求管理

步骤：

定义需求基线

处理需求变更

需求跟踪

需求开发是主线、目标，需求管理师辅助、支持与保障

9.6.1 需求开发

1.需求开发的工作

问题识别

分析与综合

贬值需求规格说明书

需求分析与评审

2.需求开发的原则

必须表达和理解问题的信息域和功能域

必须表示软件的行为

必须划分秒速信息、功能、和行为模型

分析过程应该从要素到细节实现

必须按照自顶向下分析问题不断细化

要给出系统的逻辑和物理视图

3.需求分类

功能需求

非公能需求

设计约束

业务需求：描述系统业务的需求就是问题定义，面对业务专家

用户需求：用户角度描述的需求，面对用户

系统需求：开发角度的需求，面对开发人员

9.6.2 需求获取

1.要捕获的信息

问题域相关信息（业务资料、组织机构、业务流程）；要解决的问题信息；用户的需求与约束

2.信息的来源

方法：列出一个问题域来源的对照表

可能来源：客户、原有系统、原有系统用户、竞争对手的产品、领域专家、技术法规与标准

3.需求捕获技术

1) 用户访谈，找关键用户进行面对面访谈，访谈前要做好准备，列举可能问题，有目标、有中心的访谈

- 2) 用户调查，调查问卷的方式
- 3) 现场观摩，到用户现场观摩实际的业务操作及工作流程
- 4) 阅读历史稳当（文档考古），查阅相关历史资料
- 5) 联合讨论会，关键客户代表、分析员、开发代表一起讨论，是有效的但高成本的方式

联合讨论会步骤：

- a.互相认识
- b.讨论原系统的不足
- c.对新系统的期望
- d.列出清单及会议纪要

4.需求捕获的策略

采用迭代多次反复完善的策略

9.6.3 需求分析

方法：结构化分析方法；面上对象分析方法；面向问题域的分析（PDOA）

1.业务流程分析

步骤：

- 1) 通过调查掌握基本情况
- 2) 描述现有业务流程（绘制业务流程图）
- 3) 确认现有业务流程
- 4) 分析业务流程
- 5) 发现问题，提出解决方案
- 6) 提出优化后的业务流程

2.数据流图 DFD

作用：

是表达用户需求的工具，系统分析的一种手段

是分析结果的表达方式，是系统设计的重要材料

是存档的文字材料

基本符号：

数据流、加工、数据存储、外部实体

层次图：

可以分成多层，上层是下层的父层

绘制 DFD 步骤：

自顶向下，逐层分解

绘制 DFD 原则：

- 1)只限于基本符号
- 2) 顶层 DFD必须包括全部基本元素

- 3) 顶层数据流必须封闭在外部实体之间
- 4) 每个加工至少有一个输入流或输出流
- 5) 按层给加工框编号
- 6) 子图的数据流向要与父图一致
- 7) 可以加入物质流
- 8) 每个元素必须有名字
- 9) 不可加入控制流

3.数据字典

数据字典是对 DFD 中的元素的定义集合，主要是对加工的描述，对加工的描述方法有结构化语言、判定树、判定表

9.6.4 需求定义

1.严格定义方法

使用该方法的前提：

- 1) 所有需求都能被预定义
- 2) 开发人员与用户之间能够准确而清晰的沟通
- 3) 采用图形 / 文字可以充分体现最终系统

缺陷：

- 1) 文档量大
- 2) 开发过程可见性差，用户反馈延迟大

2.原型法

作用：

可直观的描述难以理解的需求

很好的降低交流障碍

直观感性的反应需求

实用

3.软件需求说明书 SRS

软件需求说明书面向开发人员

编写方法：

- 1) 结构化自然语言编写文本型稳当
- 2) 建立图形化模型
- 3) 编写形式化规格说明

9.6.5 需求管理

3 方面工作：

需求基线：基线是基准，以基线为衡量标准

需求变更：管理需求变更

需求跟踪目的：

审核，确保每个需求都被应用

修改需求的所有相关元素的更新

跟踪变更过程及实施

跟踪需求实现过程的状态

再工程

新旧功能的比对与借鉴

跟踪每个开发人员与功能的对照关系，减少人员离开时带来的风险

测试出错时定位最优可能的代码段

需求跟踪工具：需求跟踪矩阵，用于对照功能与需求的关系，用于产品到需求的双向跟踪

9.7 软件设计

主要任务：

1.制定设计规范标准

2.完成总体设计包括模块划分、模块功能、模块间依赖关系、模块组成关系

3.设计处理方式包括算法、性能、周期、响应时间吞吐率、精度等

4.设计数据结构

5.可靠性设计

6.编写设计文档包括概要设计、详细设计、数据库设计、用户手册和初步测试计划

7.设计评审，对设计文档进行评审

9.7.1 软件设计活动

1.数据设计

最先进行数据设计， E-R图实体关系设计

2.软件结构设计

模块设计

3.人机界面设计（接口设计）

各种接口设计，模块间的接口、用户接口、外部实体接口

4.过程设计

最后进行过程设计，由顶进入，从低端退出，读者可以很容易理解过程流

9.7.2 结构化设计

是面向数据流的设计方法，是自顶向下逐步细化及模块化的设计，它包括系统结构设计、接口设计、数据设计、过程设计。

设计原则：

- 1.抽象化，对各个层次进行抽象化设计
- 2.自顶向下，逐步求精，从顶层开始设计也就是模块化设计，然后逐步细化到模块功能设计，先概要设计，再详细设计
- 3.信息隐蔽，对外数据交互采用接口方式，不要直接调用内部方法，要隐蔽内部方法，对外通过接口访问
- 4.模块独立性，高内聚，低耦合

9.7.3 workflow 设计

workflow 管理系统功能（WFMS）

流程定义

创建流程实例

监控管理实例

1.workflow 与 BPR

workflow 管理系统主要用户是和企业业务流程重组技术相结合管理企业流程定义

2.workflow 机

流程引擎，用于对流程定义进行解释，并实现流程实例的生成、激活、挂起、终止等，实现流程转换包括串行、并行、汇聚等操作

3.实现 ERP 与 OA 的集成

workflow 可以实现 ERP 系统与 OA 系统流程集成

9.8 软件测试

软件测试是软件质量保证的主要手段之一，是在软件提供用户使用之前必须完成的步骤，测试过程中要尽可能多的发现问题

测试是对软件的度量，测试的作用：

- 1.软件测试是为了发现错误而执行的过程
- 2.测试是为了证明程序有错，而非证明无错
- 3.一个好的测试用例，是发现了至今未被发现的错误
- 4.一个好的测试是发现了至今未被发现的错误的测试

9.8.1 测试的类型

1.动态测试

黑盒法，不考虑软件内部逻辑，主要方法有等价类划分、边界值分析、错误推测、因果图法

白盒法，面向程序实现测试，对程序内部逻辑进行测试

灰盒法，介于白盒黑盒之间的测试

2.静态测试

桌前检查，程序员进行单元测试之前对代码进行分析检验

代码审查，开发与测试人员共同对程序进行分析讨论的过程

代码走查

9.8.2 测试的阶段

测试类型：

1.单元测试，使用白盒测试方法，测试模块功能代码的逻辑性以及根据详细设计说明书进行测试

2.集成测试，将通过单元测试的模块进行集成然后测试集成功能，需要辅助模块驱动模块（模拟主模块），桩模块（模拟子模块），是一个循序渐进的过程，不断的集成不断的测试

3.确认测试，分为内部确认测试（开发组进行的测试，根据 SRS），Alpha 测试（内侧），Beta 测试（公测）

4.系统测试，整体环境的测试包括软件包，硬件，网络环境等综合测试，包括功能测试、性能测试、安全性测试、实施测试、用户界面测试等，其测试计划是在系统分析（需求分析）阶段完成的

9.8.3 性能测试

通过自动化测试工具如 loadrunner 等模拟多种情况进行性能方面的测试，如负载测试、压力测试等

1.性能测试的目的

目的是是否达到用户提出的性能指标，发现系统存在的性能瓶颈，并优化软件达到优化系统的目的

具体包括以下几个方面：

1) 评估系统的能力，测试负载和响应时间

2) 识别体系中的弱点，突破负荷，也就是超负荷运作，测试系统瓶颈

3) 系统调优，根据测试出来的系统瓶颈问题进行调优

4) 检测软件中的问题，长时间运行

5) 验证稳定性和可靠性，高负荷环境下测试系统稳定性和可靠性

2.性能测试类型

- 1) 负载测试，高负载环境下的运行情况
- 2) 强度测试，低配置环境的运行情况
- 3) 容量测试，高访问量

3.性能测试的步骤

- 1) 制定目标分析系统
- 2) 选择测试度量方法
- 3) 学习相关技术和工具
- 4) 制定评估标准
- 5) 设计测试用例
- 6) 运行测试用例
- 7) 分析测试结果

4.负载压力测试

主要测试高用户访问量、高并发、大数据量、持续运行时间等方面

目的：

模拟真实环境下的系统性能，评估是否可以满足系统性能需求

预见系统负载压力承受力，对预期进行评估

进行瓶颈分析并优化

关键指标：

客户端的并发用户数、响应时间、吞吐量

网络带宽利用率、网络负载、延迟、传输错误

服务器硬件指标、数据库运行情况包括 sql 执行情况、资源竞争及死锁等

9.8.4 测试自动化

优点：

- 1.比人工测试速度快
- 2.效率高
- 3.准确性高
- 4.可连续执行 7*24 运行测试
- 5.可模拟现实环境

缺点：

- 1.不能完全替代人工测试
- 2.不能减少人力成本
- 3.不能免费获得，有学习及脚本开发成本
- 4.不能降低测试工作量

9.8.5 软件调试

调试是在测试过程之后

调试针对测试中发现的问题

调试的过程：

- 1.定位问题
- 2.解决问题
- 3.回归测试

9.8.6 测试设计

测试过程包括：测试计划、测试设计、执行测试、测试评估

测试设计包括：用户层测试设计、应用层测试设计、功能层测试设计、子系统层测试设计、协议层测试设计

9.8.7 测试管理

测试管理包括：

- 1.测试团队管理
- 2.测试计划管理，测试计划从需求分析阶段开始到软件设计阶段结束时完成
- 3.缺陷跟踪管理，缺陷的提交，跟踪修复情况，处理缺陷，关闭
- 4.测试件管理，测试件包括经验教训、测试技巧、测试工具、规格文档、测试脚本

9.9 系统运行和维护

1.略

2.转换策略

- 1) 直接转换，完全替换
- 2) 逐步转换，部分功能替换
- 3) 并行转换， 2 个系统同时使用一段时间，业务数据在新旧系统中操作 2 次

9.9.2 遗留系统的处理策略

1.遗留系统的评价方法

启动平价

商业价值评价

外部环境评价

应用软件评价
分析评价结果

2.遗留系统的演化策略

淘汰策略：完全没有利用价值，淘汰重新开发

继承策略：有部分利用价值，以旧系统为基础，继承旧系统的设计

改造策略：在旧系统的基础上进行改造

集成策略：功能可用但不能覆盖全部业务，需要再增加系统与旧系统进行集成

9.9.3 软件维护

维护工作占整个软件的生命周期的 60%-80%

类型：就是鱼丸

纠错性维护 /改正性维护： 20%，发现越早成本越低

适应性维护： 25%

预防性维护： 5%

完善性维护： 50%

影响维护工作量的因素：

系统大小

程序设计语言

系统年龄

数据库技术的应用

先进的软件开发技术

9.10 软件开发环境与工具

9.10.1 软件开发环境

ICASE集成化计算机辅助软件工程，根据需求规格说明书进行自动开发以及自动环境搭建的软件

，自动生成相关的稳当

9.10.2 软件开发工具

建模工具： IBM Rational rose UML；excel WinA&D;Excel QuickUML;Metamill Metamill UML

设计工具： IBM Rational rose;Sybase PowerDesigner

编程工具：

测试工具： IBM Rational TestStudio;QACenter;WinRunner;LoadRunner

项目管理工具： PMOffice ； P3E

10.面向对象方法（软件开发方法）

10.1 基本概念

- 1.对象，是类的一个实例，组成包括对象名、属性和操作
- 2.类，组成类名、属性和操作，每个类一般都有实例，抽象类不能实例化，只能被继承
- 3.继承，A 继承 B，A 是 B 的子类，B 是 A 的父类，B 是 A 的泛化，子类继承父类的属性和操作
- 4.封装，对内高度内聚，隐藏实现，对外低耦合，通过接口访问
- 5.消息，调用对象接口方法
- 6.多态性，继承后重载就是静态的多态，覆写方法就是动态的多态；还有一种多态表现是，eat（）方法在人对象与动物对象中使用的效果不同，人 .eat() != 动物 .eat()

10.2 统一建模语言 UML

10.2.1UML 的结构

- 1.构造块：事物、关系、图
- 2.公共机制：规格说明、修饰、公共分类、扩展机制
- 3.视图：逻辑视图、进程视图、实现视图、部署视图、用例视图

10.2.2 事物

- 1.结构事物

类

接口

协作

用例

活动类

构件

节点

- 2.行为事物

交互

状态机

3.分组事物
包

4.注释事物
解释

10.2.3 关系

主要关系：依赖、关联、泛化、实现

1.用例之间的关系
包含《 include 》或《 use 》
扩展《 extend 》
泛化

2.类之间的关系
关联关系：直线表示
依赖关系：虚线实心箭头
泛化关系：直线空心箭头
聚合关系：直线空心菱形
组合关系：直线实心菱形
实现关系：虚线空心箭头
流关系：虚线箭头

10.2.4 图形

1.类图
2.对象图
3.构件图
4.组合结构图
5.用例图
6.顺序图
7.通信图
8.计时图
9.状态图
10.活动图
11.部署图
12.制品图
13.包图

14. 交互概览图

10.3 面向对象分析 OOA

10.3.1 用例模型

建立用例模型的步骤

1. 识别参与者：用户角色
2. 合并需求获得用例：功能
3. 细化用例描述：用例流程，角色如何使用功能

10.3.2 分析模型

分析步骤

1. 发现领域对象：定义概念类，一般每个用例对应一个概念类
2. 识别对象的属性：名称，解释，数据类型等
3. 识别对象关系：泛化，关联关系等
4. 为类添加职责：方法或操作，以及状态，状态可用状态图描述
5. 建立交互图：顺序图、交互概览图、通信图、定时图

10.4 面向对象设计 OOD

分类：系统设计（概要设计）；对象设计（详细设计）

面向对象设计基本准则：模块化、抽象、信息隐蔽、高内聚低耦合

具体原则：

1. 单一职责原则：一个模块只负责一个职责，避免一个模块负责多个职责出现“杂凑类”
2. 开放-封闭原则：对于扩展开放，对于修改封闭；通过调用不同的接口来实现不同的需求，而不需要每次都修改内部逻辑
3. 李氏替换原则：基类 / 父类要通用抽象，子类要扩展父类
4. 依赖倒置原则：上层模块不能依赖下层模块
5. 接口隔离原则：一个接口实现一个业务，尽量不要使用一个大的接口要使用多个接口实现多个业务
6. 组合重用原则：尽量使用组合来替代继承
7. 迪米特原则：又叫最少知识法则，就是模块尽量独立，最少依赖其他模块，面向接口编程而非实现

10.5 面向对象测试

优点：

- 1.介入较早可在需求确定之前进行，可以帮助分析需求
- 2.早期介入可降低成本，由于可以尽早发现错误
- 3.更早的编写测试用例，可以是开发与测试人员对需求理解一致
- 4.注重软件实质

测试的 4 个层次：

算法层

类层

模板层

系统层

10.6 设计模式

定义：是一套被反复使用、多数人知晓的、经过分类编目的、设计总结与经验，使用设计模式为了可重用代码设计以及体系结构。

10.6.1 设计模式的组成

- 1.模式名称，模式名称
- 2.问题，什么时候使用该模式
- 3.解决方案，设计的基本要素等
- 4.效果，使用后的结果与衡量

10.6.2 设计模式的分类

创建型（对象的创建）

- 1.抽象工厂 AbstractFactory：抽象工厂类，子类实现抽象工厂类，不同的子类不同的实现
- 2.生成器 Builder：抽象类中包含多个方法，子类需要继承并实现，不同子类不用实现
- 3.工厂方法 FactoryMethod：通过一个创建对象的接口，根据子类决定输出的对象
- 4.原型模式 Prototype：利用克隆方法
- 5.单例模式 Singleton：只能有一个实例

结构型（对象组合）：就是引入接口，然后传入实现类

- 1.适配器模式 Adapter: 将原接口方法通过适配器用其他接口方法实现
- 2.桥梁模式 Bridge：A a=new A(); Bridge b=new Bridge(a);b.method();,b 中放那个对象就执行那个对象的实现接口方法

3.组合模式 Composite：利用树型结构

4.装饰模式 Decorator :类似 sitmesh,类似桥梁模式，就是在装饰类中引用接口方法，用 get,set 方法获取实现类；以上和桥梁方法一样，下面是装饰方法就是把公共的方法集成然后各自特色方法实现在制定的位置

5.外观模式 Facade对外有一套统一的接口，也就是面向接口编程

6.享元模式 Flyweight

7.代理模式 Proxy：代理类实现公共接口并引用该接口，然后将 get,set 实现类，代理类用传入的实现类的方法实现接口方法，也就是代理类不实际实现接口方法，只是代理执行其他实现类的方法

行为型（对象间的交互，一般用 if-else 进行判断）

1.责任链模式 ChainOfResponsibility：类似工作流，定义好责任链后，通过 if-else 判断是否符合自己处理范围，不符合就推到其他责任人处理

2.命令模式 Command：不同于组合型，组合型是类的实现，而命令行是抽象类每个方法都可以独立实现，也就是传入过个类对象，每个对象实现抽象类中的一个方法

3.解释器模式 Interpreter：利用规定进行解释方法

4.迭代器模式 Iterator：自己调用自己，迭代计算，最后要有终止条件否则会死循环

5.中介者模式 Mediator

6.备忘录模式 Memento

7.观察者模式 Observer

8.状态模式 State

9.策略模式 Strategy

10.模板模式 TemplateMethod

11.访问者模式 Visitor

11.新技术应用

11.1 构件与软件重用

构件：功能相对独立的可重用价值的软件单元

11.1.1 软件重用

将已有软件及其有效成分用于构造新的软件或系统，构件在不断的被重用与完善的过程中，更加完善稳定，构件要具有可变性，根据引用系统的环境来改变构件的配置来满足新系统要求

11.1.2 构件标准

- 1.CORBA通用对象请求代理结构，由 OMG 提供
- 2.EJB企业及 JavaBean，由 SUN 提供
- 3.由微软提供
COM
DCOM 分布式构建对象模型，是 COM 的扩展
COM+，是新的产品，而非 COM 的升级产品

11.1.3 构件获取

- 1.从现有构件中获得符合要求的构件，直接使用或适应性修改
- 2.通过遗留工程获取， 通过领域工程在现有系统中进行分析提炼， 领域工程步骤： 领域分析、
领域涉及、领域实现
- 3.从市场上购买
- 4.开发构件

11.1.4 构件管理

构件组织要求：

- 1.可维护
- 2.可查询，精准与匹配查询
- 3.功能匹配查询或语法匹配查询
- 4.对构件精准描述
- 5.易用

组织方法：

- 1.关键字分类法，关键字
- 2.刻面分类法，关键特征
- 3.超文本方法，全文检索

构件分类

- 1.独立而成熟的构件
- 2.有限制的构件
- 3.适应性构件
- 4.装配的构件
- 5.可修改的构件

11.1.5 构件重用

- 1.检索与提取构件
- 2.理解与评价构件
- 3.修改构件
- 4.构件组装

11.2 中间件技术（类似适配器）

11.2.1 中间件定义

- 1.在分布式系统环境中，中间件处于操作系统之上应用程序之下
- 2.是一种独立的系统软件或服务程序，用于不同技术间共享资源
- 3.中间件是一类软件，而非一种软件
- 4.中间件不仅仅实现互连，还要实现互操作
- 5.中间件是基于分布式处理的软件，其特点是网络通信能力

11.2.2 中间件的分类

- 1.RPC, 远程过程调用， java 中通过调用 RMI（远程方法调用）
- 2.MOM，面向消息中间件，如 MQ,MessageQ
- 3.ORB是 OMG 退出的对象管理结构 OMA 的核心组件
- 4.数据库访问中间件， ODBC,JDBC

底层中间件： JVM(SUN),CLR(MS),ACE,JDBC,ODBC

通用型中间件： CORBA,J2EE,MOM,COM

集成型中间件： workflow，EAI，websphere,weblogic

11.3 J2EE 与 .NET 平台

11.3.1 J2EE核心技术

J2EE是为设计、开发、装配和不熟企业应用程序提供的一个基于组件的解决方案，提供了分布式程序模型，他提供了整套企业应用开发的解决方案包括：

服务器 tomcat，jboss 等

EJB

JDBC访问数据库用的

JMS消息组件

等等一切企业应用开发中需要的组件与规范

分布式程序模型：

- 1.客户层， web 浏览器
- 2.J2EE服务器（ web 层，业务组件层）， web 页面和应用程序
- 3.企业信息层，数据库

J2EE的规范：

- 1.J2EE平台
- 2.J2EE应用编程模型
- 3.兼容测试套件
- 4.参考实现

核心技术：

- 1.EJB企业级 javaBean，包括会话 Bean（逻辑）和实体 Bean（持久化对象）
- 2.JDBC, 数据库中间件，提供数据库访问接口，实现多种数据库的访问方言
- 3.java servlet，java 服务器端小程序
- 4.jsp，java 服务器页面，是一个高层的 servlet
- 5.JMS，java 消息服务
- 6.JNDI，java 命名目录接口，例如使用 xml 进行数据库配置的方式
- 7.JTA, java 事务 API
- 8.JMA，java mail api java 邮件 API
- 9.JAXP, java xml 解析 API
- 10.JCA连接体系， JAAS认证授权 等等

11.3.2 .NET 平台

组成：

- 1.操作系统 MS
- 2.专用服务器
- 3.web 服务器
- 4.开发工具， VisualStudio.NET
- 5..NET Framework,核心部分是 CLR(类似 java 的 JVM)

11.3.3 比较分析

J2EE: 开源，可编译成 class文件在 JVM 上运行实现跨平台，多种组件

.NET: 不开源，编译生成 .exe 和.dll 不能编译成机器码

11.4 软件即服务 SaaS

云计算包括： SaaS软件云 ,PaaS平台云 ,IaaS基础设施云

11.5 虚拟计算

网络资源的聚合与协同是虚拟计算环境的核心

11.5.1 P2P 计算

定义：对等计算，无服务端概念，各个阶段平等均是客户端也是服务端

11.5.2 云计算

11.5.3 普适计算

11.6 多核技术

11.7 片上系统 SoC 系统级芯片

12 项目管理

12.2 项目生命周期

- 1.概念阶段：包括需求收集、项目策划、可研、风险评估、项目建议书
- 2.开发阶段：要对项目任务与资源进行计划及配置，包括确定范围和目标、确定项目组成员、确定技术路线、工作分解、计划、质量保证、成本控制、风险控制、沟通管理等工作
- 3.实施阶段：按照项目计划实施项目工作，该阶段时间最长工作量最大，根据 WBS 以及网络计划来组织协调，确保项目保质按时完成，该阶段包括指导、监督、预测、控制等工作

4.结束阶段：包括财务清算、文档总结、评估验收、提交客户、项目总结评价

12.3 项目计划管理

12.3.1 项目计划的内容

- 1.工作计划
- 2.人员组织计划
- 3.设备采购计划
- 4.其他资源供应计划
- 5.变更控制计划
- 6.进度计划
- 7.成本投资计划
- 8.文件控制计划，对项目稳当的管理计划
- 9.支持计划，如培训支持、行政支持等

12.3.2 项目监督与控制

PDCA(plan、do、 check、 action) 戴明环

12.4 范围管理

范围管理计划：内容，如何根据初步范围说明书来编写详细范围说明书；如何编写 WBS 以及 WBS 字典，如何验收交付成果；如何进行变更请求批准

范围定义：范围边界，可交付成果

创建工作分解结构 **WBS**：将项目分解成可管理的工作包；工作包是最底层的单元；每一层都有共同特性；上层要素是下层要素之和；WBS 有配套的 WBS 字典用于扩展说明 WBS 中的信息如编码，费用，进度资源等

范围确认

范围控制：变更控制需要使变化的因素向有利的方向发展；判断变化范围是否已经发生；应对变化采取相对措施

12.5 成本管理

成本估算（要钱）：方法包括自顶向下（估算总的成本然后分配到下级单元中），自底向上（汇总），差别估算法（类比法）

成本预算（花钱）：就是将成本估算的结果分配到项目的各项具体工作上

成本控制

成本预算技术：管理储备

预算步骤：

- 1.将成本估算的总成本分摊到 WBS 的各个工作包上
- 2.将工作包分配到的成本在分配到工作包中的各项活动中
- 3.制定预算支出时间计划也就是成本预算计划，什么时候用什么资源用多少

直接成本与间接成本：直接受益是直接成本；间接受益是间接成本

间接成本产生的一些情况：

- 1.非直接成本如租金，保险等
- 2.隐没成本，产品的预研费用等
- 3.学习曲线，学习技术的费用
- 4.项目工期缩短，增加人员加班成本
- 5.质量要求高会增加项目成本

零基准预算：所有预算均不累加，均从 0 计算

12.5.3 挣值分析

概念：是进度和成本测量技术

基本参数：PV 计划进度应产生的费用如达到一半的时候进度为 50%* 总成本；AC 完成某工作实际花费的成本；EV 完成某工作计划花费的成本

公式：

- 1.剩余工作的成本（预测）： $ETC = PV - EV + P_{\text{剩余}} * AC / EV$
- 2.SV 进度差 $= EV - PV$
- 3.CV 费用差 $= EV - AC$
- 4.CPI 成本绩效指数 $= EV / AC$ 在相同条件下的计划与实际成本的比例如同样完成 50% 工作花费的计划成本与实际成本，CPI=1 与计划进度相同， >1 进度提前， <1 进度延迟
- 5.SPI 进度绩效指数 $= EV / PV$
- 6.EAC=总成本估算，预测总成本 $= BAC * AC / EV = BAC / CPI$
- 7.BAC=总成本预算，实际总成本
- 8.非典型 CPI=1，就是说实际进度与计划相同
典型 CPI≠1，就是实际进度与计划不一致，要么提前要么滞后
- 9.ETC 完工尚需

12.6 进度管理

活动定义（根据 wbs 与 wbs 字典）

12.6.1 活动排序

- 1.前导图法 PDM 也叫单代号
FS,FF,SS,SF
- 2.箭线图法 ADM , 虚活动
最长的路径是关键路径

12.6.2 活动资源估算（用多少资源）

12.6.3 活动历时估算（活动完成需要多长时间，考虑资源情况）

- 1.LOC代码行估算工作量
- 2.德尔菲法，专家背靠背评估
- 3.类比估算，参考相似的项目的工作量
- 4.功能点估计法

12.6.4 进度计划

- 1.关键路径法 CPM
- 2.关键路径，最长路径
关键路径上的活动不能耽误，其他路径上的活动可以与关键路径上的差值进行等待，如关键路径 5 天，A 路径 3 天，那么 A 路径可以等待 2 天
关键路径上的进度压缩后可能会改变关键路径，如关键路径 6 天，A 路径 5 天，压缩后关键路径 4 天那么关键路径就变成 A 了
车床铣床问题，第一个进入车床的时间最短，最后一个出铣床的时间最短，中间根据情况进行协调
- 3.时差
- 4.费用斜率
- 5.进度压缩 ,赶工增加成本，快速跟进并行工作，降低质量，减少功能
- 6.计划评审技术 PERT
期望时间 = (乐观 + 悲观 + 4 最可能) / 6
标准差 = (悲观 - 乐观) / 6
 $\pm 68\%$
 $\pm 2 \quad 95\%$
 $\pm 3 \quad 99\%$

甘特图时标网络图

12.6.5 进度控制

- 1.分析进度偏差的影响
- 2.项目进度计划的调整
- 调整关键路径上的活动
- 非关键活动的调整
- 增减工作项目
- 调整资源分配

12.7 配置管理

12.7.1 配置管理流程

- 1.编写配置管理计划
- 2.配置标识
- 3.变更管理和配置控制
- 4.配置状态说明
- 5.配置审核
- 6.版本管理和发行管理

12.7.2 配置标识

- 1.确定配置项
- 识别配置项

步骤：

识别配置项

为每个配置项指标唯一标识代号

制定配置项属性如作者、类型、程序语言等

确定配置时间

确定拥有者及责任

填写配置管理表

CCB审批与审查

配置项命名

配置项描述

- 2.基线（里程碑）

分类：
功能基线
产品基线
分配基线

- 3.建立配置管理系统
- 4.创建基线或发行基线

12.7.3 变更管理

1.配置库

开发库，开发阶段授权人可任意读写
受控库，对修改查看权限加以控制
产品库，不可随意修改

2.变更控制

CCB变更控制委员会
流程：

- 1.变更申请
- 2.变更评估
- 3.变更决策 CCB
- 4.变更实施
- 5.变更验证
- 6.沟通存档

变更申请文档内容

- 1.变更描述
- 2.变更审批，CCB与CM进行审批和批准
- 3.变更实施的信息

12.7.4 版本管理

版本号规范

12.7.5 配置审核

12.7.6 配置状态报告

状态报告对每一项变更要进行详细记录包括发生了什么，为什么发生，谁做的，什么时候发生的，有什么影响

状态报告内容：

- 1.变更请求概要包括申请日期，请求人，状态，估计工作量，实际工作量，发行版本，变更结束日期
- 2.基线库状态
- 3.发行信息
- 4.备份信息
- 5.配置管理工具状态
- 6.配置管理培训状态

12.8 质量管理

12.8.1 质量保证

工具和技术：

质量计划

质量审计，一般用第三方进行审计

过程分析，过程改进计划

质量控制技术

基准分析

软件质量保证 SQA

12.8.2 质量控制

方法：

验证

确认，功能是否与需求一直

测试

质量特性

功能（适准用依安），可靠（错易成），可用（学解操），效率（时源），可维护（试改定分），可移植（应装一替）

12.9 人力资源管理

12.9.1 人力资源管理计划

OBS 组织分解结构

RAM 责任分配矩阵

12.9.2 组建团队

团队组建手段：

谈判协调人员

事先分派

外部采购

虚拟团队

12.9.3 团队建设

团建措施：

沟通

培训

团建活动

集中办公

适当奖励表彰

团队发展过程：

形成期

震荡期

正规期

表现期

团队建设理论：

马斯洛需求层次

赫兹伯格激励理论

麦克雷戈 XY 理论

12.9.4 团队管理

绩效考核

12.10 沟通管理

12.10.1 沟通基本原理

沟通模型：

发送者 -编码 -传播媒介 -解码 -接受者（反馈发送者附带噪声干扰因素）

沟通渠道： $n(n-1)/2$

沟通方法：正式与非正式

沟通障碍

沟通原则：

内外有别

非正式沟通有助于关系融洽

采用对方能接受的方式

沟通升级原则，与对方沟通，与对方上级沟通，与自己上级沟通，自己上级与对方上级

沟通

扫除沟通障碍

冲突管理：

问题解决

妥协

圆滑

强迫

撤退回避

召开高效的会议

制定例会制度

放弃可开可不开的会议

明确会议目的及期望结果

发布会议通知

提前发送会议资料给参会人

借助视频设备

明确会议规则制定主持人

会后总结

会议纪要

会中后勤保障

12.10.2 沟通管理流程

- 1.识别干系人
- 2.制定沟通管理计划
- 3.发布信息
- 4.管理干系人期望
- 5.报告绩效

12.11 风险管理

- 1.风险管理计划
- 2.风险识别
- 3.风险定性分析
- 4.风险定量分析
决策树，期望货币价值，灵敏度分析，蒙特卡罗（模拟统计法）
- 5.风险应对计划
消极风险应对策略：避免，转移，减轻
正向风险应对策略：开拓，分享，强大扩大扩展
- 6.风险控制

12.11.4 常见风险

- 1.项目风险，人员，工期，预算，进度，资源，用户等
- 2.技术风险，技术陈旧或新技术不稳定行因素等
- 3.商业风险，市场期望度低，竞争激烈，无销售渠道等

12.12 软件过程改进

- 1.CMM 软件过程能力成熟度模型
初始级
可重复级
已定义级
已管理级
优化级
- 2.CMMI 软件能力成熟度模型
初始级
已管理级
严格定义级

定量管理级

优化级

13 软件体系结构

13.2 软件体系结构建模

5 种模型：

- 1.结构模型
- 2.框架模型
- 3.动态模型
- 4.过程模型
- 5.功能模型

4+1 视图：

- 1.逻辑视图，描述功能需求，使用者用户
 - 2.开发视图，描述软件的组织结构与管理，使用者开发人员
 - 3.进程视图，描述系统运行流程注重于性能，使用者系统集成人员
 - 4.物理视图，描述软件与硬件映射关系，注重网络系统通信等方面，使用者实施人员
 - 5.场景，是需求的抽象，用于关联各个视图。
- 不同系统侧重点不同，关注的视图也不一样

13.3 软件体系结构风格

- 1.数据流风格：批处理系统、管道 / 过滤器等
- 2.调用返回风格：主程序 / 子程序，面向对象，层次结构
- 3.独立构件风格：进程通信；事件系统
- 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；超文本系统；黑板系统

13.3.1 经典软件系统结构风格

1.管道 / 过滤器结构

定义：多个封闭的管道的组合，一个管道的输出是另外一个管道的输入

例子：UNIX SHELL程序

特点：

良好的隐蔽性和高内聚，低耦合，管道间只是数据的输出与输入关系，各个管道处理过程不能交互

- 可以简单的看做数据处理或过滤器行为
- 支持软件重用，只要适合 2 个过滤器间数据传送
- 系统维护简单，扩展性好
- 可分析吞吐量及死锁问题
- 支持并行

缺点：

- 不能处理交互性的行为，因为管道间没有交互，管道的处理是封闭的
- 效率低，因为每个管道都要进行数据处理解析与合成加工封装操作
- 导致进程成为批处理的结构

2.面向对象风格

定义：把操作与树型封装在对象中

优点：

- 高内聚低耦合，对外开放，对内隐蔽
- 可将操作分解

缺点：

- 对象引用后对象标识修改，需要响应修改引用位置
- 同时引用同一对象，会造成影响

3.基于时间的隐式调用

定义：通过构件注册事件，通过广播方式触发事件调用

优点：

- 解决面向对象方法缺点，引用的构件更改不会影响调用者
- 便于软件重用

缺点：

- 触发式的调用不能保证调用是否有响应以及调用顺序
- 数据交换问题
- 语义必须依赖被触发事件的上下文约束

4.分层系统

定义：分成多个层次，每一层为上层服务

优点：

- 支持抽象程度递增的系统设计
- 支持功能增强
- 支持重用

缺点：

- 不是所有系统都适用，都可以划分层次
- 很难找到合适的层次抽象方法

5.仓库系统知识库

仓库系统分为传统型与黑板系统

黑板系统：由知识源、黑板数据结构、控制组成，黑板数据就是经常被改变的数据，知识元通过改变黑板数据来解决问题

6.C2 风格

定义：通过连接件绑定到一起的构件网络

13.3.2 客户 / 服务器风格 (C/S)

属于分布式系统结构

两层结构的 C/S，客户端与服务器数据库端

优点：降低使用费用，因为不需要高性能服务器，客户端运行在用户机上

缺点：

- 开发成本高

- 客户端程序设计复杂

- 信息内容和形式单一

- 用户界面风格不一

- 软件移植困难

- 软件维护和升级困难

- 新技术不能轻易应用

13.3.3 多层体系结构风格

属于分布式系统结构

三层结构的 C/S，客户端（表示层），服务器端（功能层），数据库端（数据层）

客户端只实现操作将操作数据发到服务器端

服务器端接受数据进行处理计算，与数据库端交互，然后将计算结果返回客户端展示

优点：

- 便于开发设计

- 便于升级维护，降低运维成本

- 功能层有效隔离客户端与数据库，防止黑客侵入数据库

B/S 结构：客户端浏览器结构

13.3.4 富互联网应用

1.RIA 富客户端

技术平台：

- Flash

- Flex

- Bindows,js+DHTML(动态 HTML)

- java,Applet

- Ajax

- Laszlo 开源 RIA 开发环境

- XUL 基于 XML 的用户界面语言

- Avalon 是 vista 的一部分

13.4 软件体系结构评估

评估关键字：敏感点（功能构件），权衡点（性能）

- 1.调查问卷评估方法，主观
- 2.基于场景评估方法，半主观半客观
- 3.基于度量评估方法，客观

13.4.2 ATAM 评估方法

步骤：

- 1.描述 ATAM 方法，评估小组负责人讲解
- 2.描述业务动机，项目经理讲解
- 3.描述体系结构，首席设计师与设计小组讲解
- 4.确定体系结构方法
- 5.生成质量属性效用树，评估小组，设计小组，管理人员，客户代表共同进行
- 6.分析体系结构方法
- 7.讨论和对场景分级
- 8.如果有问题继续第六步，如果没有问题该步骤测试
- 9.描述评估结果

ATAM 评估的阶段

第一阶段以体系结构为中心，评估小组收集信息

第二阶段以项目干系人中心，获取干系人的观点，验证一阶段结果

13.4.3 SAAM 评估方法

步骤：

- 1.形成场景
- 2.描述体系结构
- 3.对场景进行分类和确定优先级
- 4.对间接场景的单个评估
- 5.评估场景的相互作用
- 6.形成总体评估

13.5 分布式系统设计

13.5.1 分布式系统设计的方式

- 1.基于实例的协作，最好在同构网络规模小
- 2.基于服务的协作，可适应异构以及网络规模大

13.5.2 基于 web 的分布式系统设计

用户层：浏览器，HTML/XML/JS

逻辑层：web 服务器

数据层：文件系统，数据库系统

13.6 软件产品线

定义：是一个产品集合，在公共核心资源上开发（如统一开发平台框架），可管理，满足选定的市场领域业务需求

产品线开发技术特点：过程驱动，特定领域，技术支持，以体系结构为中心

13.6.1 产品线的过程模型

1.双生命周期模型

领域工程阶段：分析领域通用功能，开发成通用的构件也就是核心构件资源，输出产品线的核心资源，提供给产品开发

领域分析，利用现有系统及资源建立领域模

领域设计，根据领域模型进行产品线设计体系结构

领域实现，根据领域设计开发领域可用资源（构件、文档、代码生成器）

应用工程：开发产品，以及特殊领域需求的构件开发，最终输出产品

需求分析，将系统需求与领域需求比较，划分为领域公共需求和独特需求

系统设计，在领域体系结构基础上，结合系统独特需求设计应用的软件体系结构

系统实现，遵照应用体系结构，用领域可重用资源实现领域公共需求，用定制开发的构件满足系统独特需求来构建新系统

总结：首先建立产品构件库，要有基础开发平台，产品基础核心要统一，特殊领域的需求通过开发特殊构件来实现特殊需求

领域工程就是开发核心资源库，可通过对现有产品进行逆向工程得到

2.SEI模型

组成部分：

核心资源开发（领域工程）：开发成果放入核心资源库

产品开发（应用工程）：使用核心资源开发产品，特殊化需求，单独开发

管理：对整个过程进行管理

主要特点：

各个过程循环交错，核心资源开发不断的开发新的核心资源，产品开发不断的使用核心资源

核心资源开发和产品开发没有先后之分

管理活动协调整个产品线开发过程的各个活动，对产品线的成败负责

三个活动均为双向互动

3.三生命周期模型

企业工程：操作计划，市场计划，企业体系结构计划

领域工程 /产品线工程：产品线确认，领域分析市场分析，架构开发，基础设施开发

资源管理：管理领域工程与产品线工程的生产的资源

领域工程 /产品线工程：商业市场分析与计划，系统需求推理，系统设计配置，系统成分

13.6.2 产品线的组织结构

产品线组织结构分为核心资源小组和产品小组，这是独立系统开发与产品线开发的主要区别

1.独立核心资源小组

设立有独立核心资源小组的组织，但是独立的核心资源小组会有局限性，与产品需求的独立性使得核心资源的开发缺少适应性市场需求的问题

为了解决这个问题，一般采用非独立的核心资源开发小组，只设立核心资源开发的管理人员，将核心资源开发小组融入系统开发小组中

2.组织模型

开发部门：所有开发集中在一个部门，该部门即负责领域工程又负责应用工程的任务，利于沟通适用于少于 30 人的组织

商务部门：类似事业部，各个部门间协调工作共享资源，但是各个部门重视自身的业务需求，而将产品线的需求放在次要位置，适合 30~100 人的组织

领域工程部门：类似产品中心，就是建立领域工程部，用于管理开发核心资源库的开发与维护，提供核心资源给其他商务单位使用，缺点是增加领域工程部与其他商务部门的沟通成本，与解决需求冲突的成本，导致开发周期增长，适合多于 100 人的组织

层次领域工程部门：适用于巨大超复杂的产品线，可设立多个层次的领域工程部门，每个层次负责一个业务范围，缺点结构臃肿，需求响应慢

3.动态组织结构

定义：根据产品线建立的不同时期来分配人员，如产品线建立初期，核心资源少，需要投入大量人力物力集中开发核心资源，在核心资源基本完成时，可将开发中心转移到产品开发中，优点是减少成本，增加开发人员在核心资源与产品开发组中流动有利于信息交流与沟通，但

是需要良好的管理技术来支撑该结构

13.6.3 产品线的建立方式

- 1.将现有产品演化为产品线：将现有产品逐渐修改升级为产品线，周期大投入多，风险小
- 2.用软件产品线替代现有产品集：停止产品开发，通过开发产品线来逐步替代现有产品集，投入减少，风险加大
- 3.权限软件产品线的演化： 全新领域中， 根据产品需求开发产品线， 前期投入少， 风险较小，缺点是核心资源需求会影响产品需求
- 4.权限软件产品线的开发：不考虑源产品集的需求，重新根据产品线的需求进行开发，总成本少，速度快，风险大

13.7 可扩展标记语言 XML

类似 HTML 超文本标记语言， XML 是可扩展标记语言

特点：

实现不同数据的集成

使用与多种应用环境

客户端数据处理与计算

数据显示多样化

局部数据更新

组成：

Schema,XSL,XLL,DTD,CSS,XSL,Xpath,RDF,DOM,SAX

13.8 WEB 服务 webservice

13.8.1 web 服务模型

组成：

服务注册中心：链接服务请求者与提供者的纽带，用于在注册中心找到需要的 web 服务

服务请求者：使用者

服务提供者：提供 web 服务

操作：发布，查找，绑定

13.8.2web 服务协议堆栈

- 1.SOAP: 基于 XML 的简单对象协议
- 2.WSDL: webservice 描述语言
- 3.UDDI:统一描述发现集成
- 4.XML : 用于对象封装

13.8.3 webservice 体系结构的优势

- 1.高度的通用性和易用性
- 2.完全的平台语言独立性
- 3.高度的集成性
- 4.容易部署和发布

13.9 面向服务的体系结构 SOA

13.9.1SOA 概述

定义：将服务定义集合到服务总线上，应用系统通过调用服务总线上的服务实现功能

特征：

- 1.松散耦合
- 2.粗粒度服务
- 3.标准化接口：通用标准接口

13.9.2 面向服务的分析与设计

- 1.底层设计层：采用 OOA 与 OOD 的思想
- 2.体系结构层：采用 EA 企业结构的理论框架
- 3.业务组织层：采用的 BPM 规则

遵循的原则：

- 1.业务驱动服务
- 2.业务敏捷是基本的业务需求

13.9.3 web 服务实现 SOA

- 1.底层传输层： HTTP,JMS,SMTP协议来实现消息传输
- 2.服务通信协议层：消息传递的技术标准，一般使用 SOAP协议
- 3.服务描述层：采用 WSDL来描述服务接口与消息
- 4.服务层：发布的 WSDL接口描述被定位和调用
- 5.业务流程层：服务调用的流程，采用 WS-BPEL(web 服务业务流程执行语言)
- 6.服务注册层：利用 UDDI 标准注册提供的服务，便于使用者查找

13.10 企业服务总线 ESB

- 1.ESB概念：是传统的中间件技术与 XML、 Webservice 等技术结合的产物
- 2.ESB优点：
 - 扩展的、基于标准的链接
 - 灵活的、服务向导的应用组合
 - 提高重用率、降低成本
 - 减少市场反应时间、提高生产率

14 安全性知识

14.1 信息系统安全体系

- 1.机密性：确保信息不暴露给未授权实体
- 2.完整性：只有得到允许的人才可以修改数据，并且可以判别数据是否已经被修改
- 3.可用性：得到授权的实体在需要时可以访问数据
- 4.可控性：可以控制授权范围内的信息流与行为
- 5.可审查性：可审查安全问题

14.1.1 安全系统体系结构

1.安全服务

认证服务：登录名密码方式或数字签名等方法

访问控制：防止未授权的访问，只有授权的资源才可以访问，如盗链

数据机密性服务：数据加密，防止盗用者截取信息后也无法解密数据

数据完整性服务：防止对未授权的数据进行破坏、修改、篡改等

不可否认服务：防止对已提交修改的数据进行否认，手段有数字签名等

2.特定的安全机制

加密机制：为受保护的数据加密

数字签名机制：用于数据完整性及不可否认性的一种手段

访问控制机制：防止未授权的请求受保护数据，如盗链

数据完整性机制：用于保护数据免受未经授权的修改，如通过对消息进行散列函数计算，进行数字签名

认证交换机制：交换标识使双方信任，如用户名密码

流量填充机制：分析数据流量的变化推出一些有用的信息线索

路由控制机制：通过路由策略，将访问指定一条路径，该路径上均为可信任节点

公正机制：第三方认证

3. 普遍性的安全机制

可信功能度

安全标记

事件检测

安全审计跟踪

安全恢复

14.1.2 安全保护等级

国标规定的 5 个安全等级

1. 用户自主保护级：通过用户与资源隔离来实现，如使用杀毒软件，适用于普通内联网用户

2. 系统审计保护级：登录规程，审计安全性和隔离资源，用户对自己的行为负责，适用于内联网或国际网商务活动，需要保护的非重要单位

3. 安全标记保护级：带有系统审计保护功能，适用于地方级国家机关、金融机构、有点通信、能源水利、交通、重点工程建设等单位

4. 结构化保护级：系统建立在一个结构化的安全策略模型之上，适用于中央级国家机关、广播电视、重要储备单位、重点机构、国防建设等部门

5. 访问验证保护级：对于访问进行监控，适用于国防关键部门，需要对信息系统进行特殊隔离的单位

14.1.3 信息安全保障体系

1. MIS+S: 初级安全保障系统，硬件软件通用，不带密码

2. S-MIS: 标准安全保障系统，建立在 PKI/CA 安全保障系统之上，软硬件通用，PKI/CA 必须带密码

3. S2-MIS: 超安全保障系统，软硬件专用，PKI/CA 必须带密码，主要的硬件和系统需要 PKI/CA 认证

14.1.4 可信计算机系统

- 1.D 类安全等级：只包括 D1 级别，该安全等级最低，只提供文件和用户安全保护
- 2.C 类安全等级：包括 C1、C2 两个级别，提供用户责任审计能力，C1 级别低于 C2，C2 级别带有审慎控制功能，C2 包含 C1 的全部安全特性
- 3.B 类安全等级：包括 B1、B2、B3 三个级别， $B1 < B2 < B3$ 等级
- 4.A 类安全等级：最高级别只包含 A1 一个级别，A1 与 B3 类似

14.2 数据安全与保密

密码体制建立在 3 个基本假设之上

- 1.随机性假设
- 2.计算假设
- 3.物理假设

14.2.1 加密体制

- 1.对称密码体制：加密和解密采用相同密钥

DES(数据加密标准)：算法密钥长度 56 位

三重 DES(双密钥)：112 位；2 个 56 位的密钥 K1, K2 加密过程：K1 加密 K2 解密再用 K1 加密，解密过程：K1 解密 K2 加密再用 K1 解密，长度加倍

IDEA：密钥长度 128 位，是在 DES 基础上发展起来的，类似三重 DES

- 2.非对称密码体制：采用公钥私钥方式，加密解密采用 2 套机制

RSA：长度 512 位，速度慢，主要用在数字签名 (DS)

14.2.2 安全杂凑算法

定义：就是使用散列函数 (Hash 函数，杂凑函数) 进行数据加密，加密后返回定长的字符串叫做消息摘要 (MD)，用于数字签名，不可抵赖，用户身份验证等问题

- 1.消息摘要算法 MD5

MD5(Message Digest algorithm5)，是由 MD2、MD3、MD4 发展而来

MD5 算法：将信息以 512 位来分组，每组再分成 16 个 32 位子组，经过处理后输出 4 个 32 位分组，然后将这 4 个 32 位分组合级联后产生 128 位散列值

- 2.安全散列算法 SHA

由美国国家安全局设计，是美国政府标准

SHA-1

SHA-2(SHA-224 SHA-256 SHA-384 SHA-512)

SHA-1在 TLS,SSL,SSH,PGP等中使用，但是已经被破解

14.2.3 PKI 与数字签名

PKI 是 CA 认证的基础，由认证中心、证书库、密钥备份及恢复系统、证书作废处理及客户端证书处理系统组成

作用：

认证

机密性

完整性

抗抵赖性

14.2.4 数字信封

定义：采用类似信封的方式，采用密码技术加密只有指定的接受人才能查阅

14.2.5 PGP

基于 RSA加密体系的邮件加密软件

14.2.6 数字水印

将一些标识信息嵌入到数字载体中（如多媒体、文档、软件中），通过隐藏信息来判断信息的准确性

14.3 计算机网络安全

14.3.1 网络安全设计

设计原则：

- 1.木桶原则：就是木桶的存水取决于最短的木板，系统整体安全性取决于最薄弱的安全性能
- 2.整体性原则：信息完整性
- 3.安全性评价与平衡原则：绝对安全难以达到，所以要均衡处理
- 4.标准化与一致性原则：要符合统一标准

- 5.技术与管理相结合原则：安全不仅靠技术还要靠管理
- 6.统筹规划，分步实施原则：统一整体规划，但是分布逐步实施
- 7.等级性原则：对安全进行分级
- 8.动态发展原则：网络威胁不断发展变化，安全防御也要随着改变
- 9.易操作性原则：安全实施方案实施要简单易操作

14.3.2 单点登录 SSO

- 1.利用 Kerberos 机制：是标准网络身份认证协议，基于受信任的第三方

特点：

安全性高

透明性高

可扩展性好

- 2.外壳脚本机制

通过系统外壳激发认证脚本

14.3.3 无线设备的安全性

- 1.认证性

GSM 全球移动通信系统的认证通过 SIM 卡存储认证信息

- 2.机密性

WAP 无线应用协议使用 SSL与 WTLS(无线传输层安全)来保护

- 3.恶意代码和病毒

14.3.4 防火墙

定义：是在内外网络边界上的过滤封锁机制

- 1.网络防火墙： 是一种特殊功能的路由器， 采用报文动态过滤技术， 工作在传输层与网络层，一般与应用网关配合使用

- 2.应用级防火墙：应用网关型防火墙，采用代理服务机制，用一个网关来管理应用服务，对进出包进行过滤与检测

分类：

双穴主机网关：采用堡垒主机，该主机有双网卡，一个内网一个外网，有 2 个 IP 地址用来隔离内外网

屏蔽主机网关：增加堡垒主机以及路由器，路由器用于过滤和甄别不安全的链接

屏蔽子网网关：适用于大型网络，采用独立子网，如某子网中包含 2 个路由 1 个壁垒主机，内外网各一个过滤路由

14.3.5 入侵检测

- 1.入侵检测技术
 - 特征检测
 - 异常检测

- 2.常用检测方法
 - 特征检测
 - 统计检测
 - 专家系统

14.3.6 虚拟专用网 VPN

- 关键技术：
- 1.安全隧道技术
 - 2.加解密技术
 - 3.密钥管理技术
 - 4.身份认证技术
 - 5.访问控制技术

14.3.7 IPSec

定义：是一个工业标准网络安全协议，为 IP 网络通信提供透明的安全服务，保护 TCP/IP 通信免遭窃听和篡改，抵御网络攻击，保持易用性

14.4 电子商务安全

- 1.SSL: 传输层安全协议，用于网络传输机密文件，由 SSL记录协议、握手协议和警报协议组成
 - 过程：
 - 接通阶段
 - 密码交换阶段
 - 会谈密码阶段
 - 检验阶段
 - 客户认证阶段

结束阶段

2.SET: 安全电子交易协议，用于网上交易支付

过程：

选取商品

选择支付方式

在 SET中，消费者对订单进行数字签名，采用双重签名技术，使商家不能看到消费者账号信息

接受订单，请求银行支付认可，信息通过支付网关到收单银行，再到电子货币发行公司确认，批准交易后，返回确认信息给电商

电商发送备案认证消息给消费者

确认发货

3.HTTPS安全套接字层上的超文本传输协议

4.认证中心 CA：是电子商务体系中的核心环节，是电子交易信赖的基础，核心功能就是发放证书，管理证书

14.5 安全管理

14.5.1 安全策略

1.安全模型

Bell-LaPadule:基于机密性的访问模型

Biba：基于完整性的访问模型

Clark-Wilaon: 基于完整性的访问模型

2.安全策略的制定

过程：初始与评估阶段、制定阶段、核准阶段、发布阶段、执行阶段和维护阶段

14.5.2 安全体系

1.病毒和木马扫描

2.安全扫描：扫描漏洞

3.日志审计系统

4.安全审计

5.个人信息控制

6.安全管理制度

14.6 计算机操作安全

14.6.1 安全威胁

评估安全威胁的方法：

1. 查阅
2. 试验
3. 调查
4. 测量

14.6.2 物理安全

设备安全，介质安全，环境安全等

14.7 系统备份与恢复

14 章总结：

1. 数字签名：防止信息抵赖
2. 加密技术：防止信息窃取
3. 完整性技术：防止信息被篡改
4. 认证技术：防止信息被假冒
5. VPN: 虚拟专网，在公网上传输私有数据到私有地址
6. SSH: 传输数据进行加密，代替 telnet , ftp 等
7. SSL: 安全文件网络传输协议
8. SET: 在线支付

15 知识产权与法律法规

15.1 著作权法

1. 有合同协约的根据合同内容归属
2. 完成单位工作任务所创作的作品，称为职务作品著作权归单位所有，作者享有署名权
3. 共同创作，归创作者共同所有
4. 委托创作，著作权归受托人所有

15.1.1 著作权法主体

著作权人：原始作者
受让者：后继著作权人，非作者，通过转移活动享有

15.1.2 著作权

享有的权利：

- 1.发表权
- 2.署名权
- 3.修改权
- 4.保护作品完整全
- 5.使用权

一些时间期限：

著作权属于公民的：

- 1.署名权、修改权、保护作品完整权没有时间限制
- 2.发表权、使用权、获得报酬权截止于作者终生以及死亡后的 50 年的 12 月 31 日。

著作权属于单位的：

- 1.发表权、使用权和获得报酬权保护期为 50 年的 12 月 31 日
- 2.50 年内未发表的，不予以保护
- 3.有变更的按照变更内容进行分配权限

第三方使用需要得到著作权人的授权，合同期限不超过 10 年，超过可续签

15.2 计算机软件保护条例

15.2.2 软件著作权

权利内容：

- 发表权
- 署名权
- 修改权
- 复制权
- 发行权
- 出租权
- 信息网络传播权
- 翻译权
- 使用许可权

获得报酬权
转让权

15.3 招标法

15.3.1 招标

一、招标代理机构

可委托代理机构办理招标事宜，也可自行招标

二、招标公告

标注招标人的名称地址项目性质数量实施地点等信息

公开招标，通过媒体渠道发布

邀请招标，给三个以上的满足条件的投标人发送表述

三、招标文件

包含合同条款、标段、工期、权益责任、评标标准及报价等

15.3.2 投标

15.3.3 评标

1.开标

2.评标

3.中标

4.分包

15.3.4 关键问题

必须进行招标的项目

1.大型基础设施，公共事业等关系社会公共利益，公众安全的项目

2.全部或部分使用国有资金投资或者国家融资项目

3.使用国际组织或者外国政府贷款援助的项目

分类

邀请招标，机密项目或特殊部分可以使用邀请招标

公开招标

招标前要对已购买表述的用户信息及数量保密

招标发标书或公告到开标不得少于 20 日

开标到投标截止时间不能少于 20 天

修改招标文件的时间到截止时间不能少于 15 天

少于三人投标需要重新招标

招标人收到表述不可查看内容，应当签收存档

标书只有按时送达才可签收否则拒收

竞标价不得少于成本价

提交标书截止时间即开标时间，不得延迟

评标委员会，人数 5 人以上专家不的少于三分之二

中标标准，满足条件价格最低不得少于成本价

中标后 30 日内签合同

中标后 15 日内中标人提交书面报告

联合投标资质为联合投标人资质最低的为准

分包条件：甲方同意、分包人要有响应的条件、只能分包非主体、分包人不可再次分包

投标保证金不得超过项目预算 2%

资格预审文件或招标文件的发售期不得少于 5 天

提交资格预审申请文件的时间自停售值日起不得少于 5 天

评标结束后 3 天内公布结果

标底，价格低价要保密

专利权：

发明：保护期限 20 年

使用新型：10 年

外观设计：10 年

商标：
注册商标有效期 10 年，期满前 6 个月内申请续展注册，未注册的可给予 6 个月的宽展期，再次过期则注销，续展期也为 10 年

16 标准化知识

16.1 标准化基础

16.1.1 标准的制定

- 1.标准的层次
 - 国际标准
 - 国家标准
 - 行业标准
 - 地方标准
 - 企业标准
- 2.标准的分类
 - 强制性标准与推荐性标准

16.1.2 标准的表示

中国国家标准代号
GB:国标强制
GB/T:国标推荐
GB/Z:国标指导

行业标准代号
CY新闻出版
DA 档案
DL 电力
GA 公安
GY广影
HB 航空
HJ环境
JC建材
JG建工
LD 劳动

SJ电子
WH 文化
YD 邮电
YZ 邮政

GJB国家军用标准，属于行业标准
GSB国际实物标准
DB 地方标准

16.1.3 ISO9000 标准族

国际标准化组织中的质量管理和质量保证委员会定制的一系列标准
ISO认证证书有效期 3 年，第一次有效期内每年检查两次，三年期满换证后每年检查一次

16.2 文档标准

16.2.1 GB/T16680-1996 软件文档管理指南

文档作用：

- 1.管理依据
- 2.任务之间联系的凭证
- 3.质量保证
- 4.培训与参考
- 5.软件维护支持
- 6.历史档案

文档类型：

- 1.开发文档：软件需求、设计、测试、质量保证的文档
- 2.产品文档：用户指南、参考手册、宣传资料、产品白皮书、产品手册、培训手册
- 3.管理文档：进度计划、变更信息、过程文档、责任定义等

文档计划： 是指一个描述文档编写工作的管理用文档， 主要内容包括何时由谁如何编写文档

计划详细内容：

应编制文档目录

参考标准

指定文档管理员

提供编制文档条件，落实负责人员，编制经费及工具

明确保证文档质量的方法，如评审及鉴定等

绘制进度表，进度计划

文档等级：

- 1.最低限度文档（ 1 级）：开发工作量低于 1 人月的自用程序，包括程序清单、开发记录、测试数据和程序简介
- 2.内部文档（ 2 级）：专用程序，可与其他用户共享
- 3.工作文档（ 3 级）：同一单位内若干人联合开发的程序，可给其他单位使用的程序
- 4.正式文档（ 4 级）：正式发行的软件产品，关键性程序如财务软件等

16.2.2 GB/T8567-2006 计算机软件文档编制规范

规定了 6 个阶段需要编制的文档及内容，还规定了面向兑现个软件编制文档

面向对象文档：

- 1.总体说明文档
- 2.用例图文档
- 3.类图文档
- 4.顺序图文档
- 5.协作图文档
- 6.状态图文档
- 7.活动图文档
- 8.构件图文档
- 9.部署图文档
- 10.包图文档

6 阶段的软件文档：

- 1.可行性研究
可行性研究报告
项目开发计划

- 2.需求分析
软件需求规格说明书
数据库要求说明书
用户手册
开发进度月报
测试计划

- 3.软件测试
概要设计说明书
详细设计说明书
数据库设计说明书
用户手册
操作手册

开发进度月报
测试计划

4.编码自测
用户手册
操作手册
模块开发卷宗
开发进度月报

5.集成测试
模块开发卷宗
开发进度月报
测试分析报告
项目开发总结

6.运行维护
维护报告

文档控制
由于各个文档是在项目进行中不断的完善与修改的，因此要保证文档与程序的一致性以及文档间的一致性所要做的控制工作

16.2.3 GB/T 9385-1988 计算机软件需求说明编制指南 SRS

1988 年 12 月 1 日起实施

SRS特性：
无歧义性
完整性
可验证性
一致性
可修改性
可追踪性
运行和维护阶段的可使用性

16.2.4 GB/T 16260-2006 软件产品质量标准

功能性
 适合性
 准确性
 互用性
 依从性

- 安全性
- 可靠性
 - 容错性
 - 易恢复性
 - 成熟性
- 可/易用性
 - 易学性
 - 易理解性
 - 易操作性
- 效率性
 - 时间特性
 - 资源特性
- 可维护性
 - 可测试性
 - 可修改性
 - 稳定性
 - 易分析性
- 可移植性
 - 适应性
 - 易安装性
 - 一致性
 - 可替换性

17 计算题

1. 风险管理

决策树和期望货币价值 EMV

$EMV = \text{比例}_1 * \text{方案}_1 + \text{比例}_2 * \text{方案}_2 + \dots + \text{比例}_n * \text{方案}_n$

2. 采购管理

加权系统：比重 $_1 * \text{评分}_1 + \text{比重}_2 * \text{评分}_2 + \dots$

自制与外购决策：采用决策树也就是计算 EMV

投资回收期：

贴现率就是利率

静态回收期不考虑时间价值也就是利率

第一年动态价值 = 第一年投资 / (1 + 利率)

第二年动态价值 = 第二年投资 / (1 + 利率)²

第 N 年动态价值 = 第 N 年投资 / (1 + 利率)^N ?

回收期 = 收回成本的时间

回收率 = 1 / 回收期

年份编号从 0 开始，所以当年动态投资金额 = 投资 / (1 + 利率)⁰ = 当年投资金额 / 1

3. 沟通管理
沟通渠道 $=n(n-1)/2$
 4. 盈亏平衡点
收回成本的时间
 5. 进度网络计算 PERT(计划评审技术)
期望时间 $= (悲观 + 乐观 + 4 * 最可能) / 6$
标准差 $= (悲观 - 乐观) / 6$
三点估算：
期望时间 \pm $=68\%$
期望时间 ± 2 $=95\%$
期望时间 ± 3 $=99\%$
 6. 6 标时
E(early)、F(finish)、L(last)、S(start)
ES EF 最早开始，最早结束
LS LF 最迟开始，最迟结束
总时差 $=LS-ES=LF-EF$
自由时差 $=EF$ 紧后活动的 ES
最早时间正推取大的
最迟时间倒退取小的
最长的路径就是关键路径，关键路径的时间就是项目的总工期
关键路径上的活动总时差与自由时差均为 0
 7. 资源平衡
 8. 改变工期要调整关键路径上的活动，但是调整关键路径上的活动会引起关键路径缩短，此时可能会更改关键路径
 9. [挣值计算](#)
-

二、网络资源考点汇总

第一章

- 1、软件的概念：软件是一种逻辑实体，而不是具体的物理实体
软件 (software) 是计算机系统中与硬件 (hardware) 相互依存的另一部分，它包括程序 (program)、相关数据 (data) 及其说明文档 (document)。
其中：
程序 -- 按事先设计的功能和性能要求执行的指令序列；
数据 -- 使程序能正常操纵信息的数据结构；
文档 -- 与程序开发、维护和使用有关的图文材料。
- 2、软件的特点：软件的生产与硬件不同；软件没有明显的制造过程。一旦研制开发成功，就可以大量拷贝同一内容的副本。软件对硬件和环境有着不同程度的依赖性。这导致了软件移植的问题。软件的开发至今尚未完全摆脱手工作坊式的开发方式，生产效率低。软件是复

杂的，而且以后会更加复杂。软件工作牵涉到很多社会因素。软件的运行和使用期间，没有硬件那样的机械磨损，老化问题。

3、软件分类：

1. 基于软件功能：系统软件（OS，DBMS）支撑软件（各种软件开发包等）、应用软件（各种MIS系统）。

2. 基于软件工作方式：实时处理软件、分时软件、交互式软件、批处理软件。

3. 基于软件规模：微型软件、小型软件、中型软件、大型软件、甚大型软件、超大型软件。

4、系统开发生命周期：系统开发生命周期（Software Development Life Cycle，SDLC）是指这样的一个过程，包括：理解信息系统对业务需求的支持，设计系统，构建系统，以及把系统移交给用户。计划、分析、设计、实现。

各阶段的任务及结束标志

一、计划阶段是理解为什么要创建信息系统和确定项目团队将如何来开发它的基本过程。计划阶段由2个步骤组成：

1) 在项目启动期间，要确定系统给组织带来的业务价值。主要通过技术可行性、经济可行性、组织可行性分析来完成。

2) 项目批准后，进入项目管理。

二、分析：分析阶段说明此系统由谁来用，用作什么，在哪里用，以及什么时候用这些问题。在此阶段，项目团队调查现有系统，确定可改进的地方，以及开发新系统的方案。主要步骤如：

1) 开发分析策略来指导项目团队工作。

2) 收集需求。

3) 分析结果，系统方案和模型组合成系统建议书。

三、设计：设计阶段确定系统将如何运行，涉及硬件、软件和网络基础设施；将要使用的用户界面，窗口、窗体和报表；所需的专用程序、数据库和文档。具体步骤如下：

1) 创建设计策略。

2) 开发系统的基本架构设计，描述要用到的软、硬件和网络设施。

3) 开发数据库和文档规格。

4) 开发程序设计规格，定义需要编写的程序和每个程序确切要做的事情。

四、实现阶段是SDLC的最后阶段，是系统实际构建阶段。主要步骤如下：

1) 系统构建。

2) 系统安装。

3) 建立系统的支持计划。

5、系统开发方法：结构化和面向对象

结构化：(1)、瀑布式开发

每个阶段都是在前一阶段完成的基础之上才进行。

优点：系统中编程之前就已确定；项目进行期间变动不大。

缺点：编程之前需要充分的设计；需求的变动无法及时得到解决。

(2)、并行开发：在概要设计完成之后分成多个子系统，然后分别进行设计和实现，最后再组合成一个系统。

优点：提高了项目开发的效率。

缺点：子项目间可能会相互影响；项目中加入了子项目的集成。

(3)、快速应用开发（Rapid Application Development，RAD）是指结构化方法的基础上创建，用于解决结构化方法中的编程之前需要充分设计和在开发过程中需求变更无法得到及

时响应的缺点，使用 RAD, 可以使系统的部分功能更快的开发并提交给用户。

遵循 RAD的方法主要有：过程为中心，数据为中心，面向对象。

(4)、敏捷开发 (Agile Development) 是一种新兴的开发方法，它是以编程为中心，注重简化过程，强调迭代式的开发。

遵循敏捷开发的方法主要有：极限编程 (eXtreme Programming, XP), Scrum 和动态系统开发方法 (Dynamic Systems Development Method, DSDM)

开发方法的选择主要考虑以下因素：

用户需求的清晰度、技术的属性程度、系统复杂度、系统可靠性、项目的时间进度要求、项目的进度可见性。

6、文档项目文档包括所有的可交付物，有关该项目的历史记录。常见的文档：可行性研究报告；各种计划、需求分析、系统各种设计、程序代码，测试脚本，数据库脚本等，各种分析报告。

第二章需求分析

1、可行性分析：可行性分析主要用于辅助组织决定是否继续项目开发的依据，主要从技术、经济和组织三个方面进行分析，并综合成可行性研究报告，在项目启动阶段的末期交付给审定委员会。技术可行性分析，即系统可以被 IT 团队成功的设计、开发和安装运行的程度。主要从以下几个方面进行分析：用户和分析员对业务应用的熟悉程度对项目开发所用到的技术的熟悉程度所要开发的项目的规模系统与其他系统的兼容性。经济可行性分析，确定与项目相关的财务风险，确定是否值得开发新系统。经济可行性分析步骤如下：确定花费和收益给花费和收益指定数值定义现金流估算项目的经济价值：回报期，平衡点，净现值无形的费用和收益：例如客户服务的改进，社会价值的提高，企业形象的提高等。

2、系统需求是指描述创建系统的业务原因和系统预期带来的价值的文档。

系统需求包括的元素：项目发起者，业务要求，业务需求，业务价值，其他方面的要求和约束

第三章

1、根据计划阶段确定项目的规模：在一般的业务应用系统中，计划阶段花 15%时间，分析阶段 20%时间，设计阶段 35%时间，实现阶段 30%时间。因此可以根据计划阶段所花的时间来估算其他阶段所需要的时间，进而得到整个项目的估算时间。

缺点：每个项目都是其特殊性，并不与这个典型的时间分配一致。

根据功能点：基于功能点估算项目规模，需要先估算出项目所需的代码行，根据代码行估算所需的时间。

2、项目工作计划是用来管理列在工作分解结构 (Work Breakdown Structure, WBS) 中的任务的一种机制，是项目经理管理项目的主要工具。在项目工作计划中还体现了：里程碑任务期限当前任务状态任务相关性 里程碑

3、wbs 项目管理使用的工具：

1) 甘特图可以体现如下信息：任务所需时间、人力资源分配、任务的先后时间关系、任务的提早或延迟。

2) PERT 图：网状图形工具，与甘特图不同，PERT很好的体现了任务之间的依赖关系，PERT还可以计算出项目的关键路径和关键事件。

3) 估算求精。

4、团队的构建（掌握）：使用合理的人员激励方式：慎用金钱激励；使用内在激励；认可；成就；工作本身的吸引力；责任感；工作晋升；新技术的学习机会。

避免冲突的策略：清晰定义项目计划；确保项目团队理解该项目对某个组织机构的重要性；创建详细的操作流程并与项目成员进行沟通；创建项目章程；创建超前的进度计划；预测项目的其他优势和可能的影。

5、了解 CASE工具：计算机辅助软件工程（CASE）是一种自动生成全部或部分开发进程的策略软件。使用 CASE可以：更好的完成和转换任务；集中开发信息并可图形化呈现；减少维护费用；提高软件性能和强化规则。

第四章

重点：

1、需求的概念：需求就是陈述系统必须要做的事或者系统必需具备的特征。

1) 需求分为功能需求和非功能需求。功能需求与系统必须执行的过程或必须包含的信息有直接关系。非功能需求指的是系统必须具备的行为属性，如性能、安全性和可用性等。功能需求：面向过程 - 系统必须执行的过程和完成的任务；面向信息 - 系统必须包含的信息；

2) 在分析阶段，从业务员角度出发，关注的是业务用户的要求，因此也称为业务需求（用户需求）；在设计阶段，主要从开发人员角度出发描述，称为系统需求。特征，及其收集需求的技术的优缺点；

2、需求的特征：描述系统必须实现的功能；描述系统必需具备的特征；在分析阶段关注用户的要求；需求在项目的不同阶段会发生改变。

典型的非功能性需求：

1) 性能表示为满足用户的需要而要求系统展示的性能 -- 吞吐量、响应时间；

2) 信息表示有关用户的信息，形式为内容、时间性、正确性和格式 -- 需要输入和输出是什么，数据存储在哪里等，对外接口；

3) 经济表示系统对减少开支或增加收益的需要 - 必须减少开支的是什么，预算限度是多少；

4) 控制表示系统必须在其中运行的环境以及必须提供的安全类型和程度 -- 访问控制，对数据的特殊处理（脱机备份等）；

5) 效率表示系统以最低成本产生输出的能力 -- 在过程中有必须消除的重复步骤，用其资源的方式中存在降低成本的方法吗；

6) 服务表示使系统可靠、灵活和可扩充的需要 - 不同类型的用户，培训、相关文档资料。

3、收集需求的技术及其优缺点：

面谈，问卷，观察现场，联合应用开发，文档分析。

1) 面谈：面谈通过直接、面对面的交互获取需求。这种方式可以用来实现以下目标：发现事实、验证事实、澄清事实、激发热情、让最终用户参与、确定需求以及征求想法和观点。

面谈的缺点：面谈耗时，费用高；面谈的成功极大地取决于采访者的个人的能力；面谈可能会由于被采访者的地理位置功；面谈有两种类型：结构化面谈和非结构化面谈。非结构化面谈的特点是涉及一般性的问题，被采访者可以引导谈话过程。这种方式容易偏离主题，采访者需要及时的引导。结构化面谈要求采访者询问一套专门设计用于从被采访者处获取特定信息的问题，根据被采访者的回答，采访者将提出额外的问题（计划的问题或非计划的问题）以进一步深入。准备是面谈成功的关键，面谈之前如果没有很好的准备，那么面谈就无法获得满意的结果。在面谈之前最好准备一份面谈指南，安排好提问的问题，每个问题的时间，

在什么地方可能需要进一步的提问等。 在面谈的过程中， 需要注意以下事项： 假定答案存在或不存在；提示线索；使用行话；显示个人偏见；谈论而不是聆听；对有关主题和被采访者的情况作出假定；使用录音 - 聆听能力差的表现。

2) 问卷：问卷调查表可以使分析员从一大群人那里收集到事实，同时保持统一的答复。尤其是当需要面对大量的人时， 其他调查研究技术都不可能有如此有效的得到结果。 问卷调查表的优点：

- 大多数的调查表可以被快速的回答；
- 调查表提供了一种可以以相对廉价的方式从大量的人中收集数据；
- 调查表可以匿名填写，因此更能得到真实的数据；回答可以快速的表格化和分析；

问卷调查表可以使分析员从一大群人那里收集到事实， 同时保持统一的答复。 尤其是当需要面对大量的人时，其他调查研究技术都不可能有如此有效的得到结果。

问卷调查表的缺点：回答者的数量经常较低；无法保证个人会回答或进一步回答所有问题；调查表往往不灵活；没有机会立即澄清对问题的含糊或不完整的回答。

制作问卷调查表：确定必须收集什么事实和观点以及你应该从谁那里收集；根据需要的事实和观点，确定是采用自由格式还是固定格式；编写问题，确保问题中没有反映个人的偏好和观点；在小样本中进行测试，发现问题及时修正；进行调查。

3) 联合应用开发： 一对一的面谈需要使用大量的时间， 使用小组会议的形式进行面谈就可以解决这个问题。 JAD就是采用小组会议的形式来获取需求。一般来说，一个成功的 JAD需要负责人、主持人、用户和管理员、记录员、参与系统开发建设的其他相关人员等角色的参与

第五章

1、重点用例分析 (10 分)DFD(10 分)、类图 (10 分)

2、建模原则 (了解) 建模的原则：选择建立什么样的模型对如何发现和解决问题具有重要的影响；每个模型可以有多种表达方式；最好的模型总是能够切合实际；孤立的模型是不完整的。任何好的系统都是由一些几乎独立的模型拼凑出来的。

UML建模 (知道) 背景：1994 年 10 月，Rational 公司的 Booch 和 Rumbaugh决定将其 Booch 方法和 OMT方法综合成一个新的建模语言，并于 1995 年 10 月公布了 Unified Method 0.8。

1995 年秋季，Jacobson 及其 OOS方法加入 Rational 公司，决定将 OOS方法与 Unified Method 进行综合，更名为 UML，并分别于 1996 年 6 月和 10 月公布了 UML 0.9 和 UML 0.91。

1996 年，DEC、HP、I-Logix、Intellicorp、IBM、ICON、MCI、Microsoft、Oracle、Rational、会，于 1997 年 1 月推出了 UML1.0，并向 OMG申请将其作为一种标准语言。1997 年 9 月产生了 UML1.1，11 月被 OMG正式采纳。1999 年 6 月，OMG发布了 UML1.3。1999 年 7 月，UMLRealTime 随 Rose RealTime 推出。2001 年 9 月，OMG发布了 UML1.4。2004 年 4 月，OMG发布了 UML 1.5。2005 年 7 月，OMG发布了 UML 2.0。

用例：描述了一系列执行的活动所产生的一些输出结果。 每个用例描述了外部用户如何来触发系统必须响应的事件。 在事件驱动模型中，系统的一切行为都被认为是对某个触发事件的响应。

相关：用例是由外部用户触发；每个用例只描述单独的任务，而不能描述多个任务；用例必须产生一个对用户有意义的结果；场景 (Scenario) 是用例的一个执行实例，是例执行；过程中的一条实际路径，一个用例可能会包含多个场景；

UML中的用例视图只能反映出两类信息：

1) 哪些外部用户会和系统发生交互

2) 系统需要实现哪些功能特性

基本信息：

1) 每个用例都有一个名称、编号和简要描述。为了明确标识某些用例在整个系统中的相对重要性，需要使用优先级来表示。用例 (use case) 是一个行为上相关的步骤序列，代表的是一个相对完整的业务任务。UML中的用例是动作步骤的集合。动作 (action) 是系统的一次执行 (能够给某个参与者输出结果值)。与参与者通信，或进行计算，或在系统内工作都可以称作动作。用例应支持多种可能发生的动作，每个动作由许多具体步骤实现。

2) 用例的元素：参与者是存在于系统之外并与系统交互的人或事。所谓“与系统交互”指的是参与者向系统发送消息，从系统中接收消息，或是在系统中交换信息。参与者是一个群体概念，代表的是一类能使用某个功能的人或事，参与者不是指某个个体。

3) 输入和输出每一个用例的主要输入、输出连同其来源和目的都要进行描述。它包括所有可能的输入输出，而不仅仅是用来通常用的部分。建造用例是一个逐渐提炼的过程，用例在分析的过程中可以逐步的完善。

4) 细节：用例需要描述详细的步骤和它们所使用到的输入和输出，这些步骤就是用例中所执行的活动。

第六章

1、活动图的概念：活动图 (activity diagram) 显示了组成复杂过程的步骤序列，如工作流或算法。活动图是对系统的行为进行建模，活动图是把系统的一项行为表示成一个可以由计算机、人或其他执行者执行的活动，通过给出活动中的各个动作以及动作之间的转移关系来描述系统的行为。

2、活动图与流程图的区别：

流程图着重描述处理过程，它的主要控制结构是顺序、分支和循环，各个处理之间有严格的顺序和时间关系；而活动图描述的则是对象活动的顺序关系所遵循的规则，它着重表现的是系统的行为，而非系统的处理过程。活动图能够表示并发活动的情形，而流程图做不到。活动图是面向对象的，而流程图是面向过程的。

3、使用活动图的目的：描述一个操作执行过程中 (操作实现的实例化) 所完成的工作 (动作)；描述对象内部的工作；显示如何执行一组相关的动作，以及这些动作如何影响它们周围的对象；显示用例的实例是如何执行动作以及如何改变对象状态；说明一次业务活动中的工人 (角色)、工作流、组织和对象是如何工作的。

4、活动图的组成：

1) 动作是构成活动的基本单位，它是原子的、不可中断的，并在动作完成后通过完成转换转向另一个状态。

动作的特点：动作是原子的，不可以分解成更小单位；动作是不可中断的；动作是瞬时完成的行为；动作可以有入转换，至少有一条出转换；动作不能有入口动作和出口动作；在一张活动图中，动作允许出现多次；

2) 活动是由一系列动作构成的，是对一项系统行为的描述。

活动的特点：活动可以分解成其他子活动或动作；活动的内部活动可以用另一个活动图来表示；活动可以有入口动作和出口动作，还可以有内部转移；

3) 动作流

4) 条件是让转移修改任何工作流的方向所必须的。

5、顺序图：顺序图 (SequenceDiagram，时序图，序列图) 详细描述对象间传送消息的时间顺序，它表示用例中的行为顺序。顺序图它详细而直观地表现了一组相互协作的对象在执

行一个 (或少量几个) 用例时的行为依赖关系， 以及操作和消息的时序关系。 类图对对象之间的消息 (交互情况) 表达不够详细 ；详细说明对消息的表达虽然详细， 但不够直观 ；顺序图既详细又直观，但通常只能表示少数几个对象之间的交互。

2) 活动对象： 活动对象可以是系统的参与者或任何有效的系统对象。 在活动图中对象的标记如下图所示。 将对象置于时序图的顶部意味着在交互开始的时候对象就已经存在了， 如果对象的位置不在顶部，那么表示对象是在交互的过程中被创建的。

生命线是一条垂直的虚线， 表示时序图中的对象在一段时间内的存在。 每个对象的底部中心的位置都带有生命线。

- 生命线是一个时间线， 从时序图的顶部一直延伸到底部， 所用的时间取决于交互持续的时间。
- 对象与生命线结合在一起称为对象的生命线， 对象的生命线包含矩形的对象图标以及图标下面的生命线。
- 如果对象在图中被创建， 那么对象符号画在创建它的消息上， 否则画在任何消息箭头上。如果对象在图中被撤销，那么用 “ x ” 表示撤销。
- 消息定义的是对象之间某种形式的通信， 它可以激发某个操作、 唤起信号或导致目标对象的创建或撤销。
- 消息是两个对象之间的单路通信，从发送方到接收方的控制信息流。
- 消息可以用于在对象间传递参数。
- 消息可以是信号，也可以是调用。

符 号	含 义	例 子
=	被定义为	
+	与	x=a+b, 则表示x由a和b组成
[,][]	或	x=[a,b]或 x=[a b], 则表示 x 由 a 或由 b 组成
{ }	重复	x={a}, 则表示 x 由 0 个或多个 a 组成
m()n	重复	x=3{a}8, 则表示 x 中至少出现 3 次 a,最多出现 8 次
()	可选	x=(a), 则表示 a 在 x 中出现, 也可不出现
...	注释	表示在两个 * 之间的内容为词条的注释
...	连接符	month=1..12 表示 month 可取 1 到 12 中任意一个值

- 在 UML中，消息使用箭头来表示，箭头的类型表示了消息的类型。

第 7 章

设计阶段的勾当主若是： （ 体味 ）

- 确定合适的系统获取策略
- 设计系统的架构
- 选择软硬件
- 设计系统人机接口
- 从逻辑模子到物理模子的转换
- 设计能完成系统过程的轨范
- 从逻辑数据模子到物理数据模子的转换
- 设计数据存储方案
- 编写最终的系统规格

常见的设计错误有：（把握）

削减设计时刻；需求伸展；过于依靠手艺和工具；在项目中心阶段改换工具。

机关新的应用系统的体例：（理解）

1. 定制开发
2. 采办软件包
3. 第三方外包开发

影响设计策略的身分

1. 营业需求

对于通用的营业需求，而且有斗劲成熟的手艺解决方案，使用定制应用轨范斗劲轻易。

对于怪异的或专用的需求，则选择开发系统。一般来说，如不美观营业需求不是公司策略的关头元素，可以选择外包。

2. 项目手艺

在项目中应用的手艺既可所以手艺性（如开发说话）的，也可所以功能性（如电子商务）的，分歧设计方案的选择取决于这些手艺在公司策略中的主要性。

3. 项目打点

定制应用轨范要求有较强的项目打点和被证实的体例论。在项目进展过程中，项目可能受到来自各个方面身分的影响，如不美观没有较强的项目打点能力则项目很可能就面临坚苦。采办和外包软件也同样需求项目打点，而且这种打点更多的来自于组织外部的沟通和交流。

4. 时刻约束

当项目存在时刻的约束时，首先应钙揭捉找一个已经成立和经由测试的系统。但操作外包来建树一个系统的时刻取决于系统和外包商的资本，如不美观处事供给者没有供给合适的处事，那么采用外包来解决所破耗的资本可能与定制开发一样。

第 8 章(综合应用)

常用的软件系统结构

1. 主机 / 终端
2. 文件 / 处事器
3. 客户 / 处事器结构 (C/S)

客户机 (Client) 和处事器 (Server) 都是自力自立的系统，它是一类按新的应用模式运行的分布式计较机系统。在这个应用模式中，用户只关心完整地解决自己的应用问题，而不关心这些应用问题由系统中哪台或哪几台计较机来完成。

4. 浏览器 / 处事器结构 (B/S)

**** 结构 (C/S 结构，B/S 结构) 的首要组成部门

· 用户默示层：接管用户的请乞降操作，将操作信息最终显示给用户。如浏览器中的 Web 页面，或者是一个 Windows 窗体轨范

- 营业逻辑层：营业逻辑层完成营业功能和逻辑。如 Web 网站中的 IIS 处事器
- 数据处事层：指后台数据库系统，也可所以各类能够存储数据的文件

5. Web Service

6. 面向处事的系统结构 (SOA)

7. SaaS (Software-as-a-Service，软件即处事) 是一种经由过程 Internet 供给软件的模式，厂商将应用软件统一部署在自己的处事器上，客户可以按照自己现实需求，经由过程互联网向厂商定购所需的应用软件处事，按定购的处事若干好多和时刻长短向厂商支出费

用，并经由过程互联网获得厂商供给的处事。

8. 软件+处事 (S+S)

9. RIA (Rich Internet Applications) 系统结构

B/S 架构的应用的几乎所有措置都集中在处事器端，造成处事器端负载过大，而且收集通信也过多。 RIA 在客户端与处事器端之间惹人被称为“客户端引擎”的中心代码层，实现客户端交互的丰硕和降低对处事器端处事的依靠。

第九、十章轨范设计

1、建树物理 DFD 的轨范：

(1) 添加实现标注（在逻辑 DFD 的基本上添加注释，即数据若何存储，数据若何流动以及过程若何实现）

(2) 绘制一小我机鸿沟（系统的任何一个过程的任何部门都不成能自动完成，所以需要确定在哪里绘制人机鸿沟）

(3) 添加系统相关的数据存储、数据流和过程（添加一些附加的数据存储、数据流和过程，这些与营业无关但可以实现一些特定的功能）

(4) 更新数据流中的数据元素（因为在前面的轨范中添加了一些数据存储、数据流和过程，是以需要对这些元素进行更新）

(5) 更新 CASE 常识库中的元数据（更新常识库中的元数据以确保与 DFD 同步。）

2、结构设计原则：

建树高内聚的模块；成立松耦合的模块；建树高扇入；避免高扇出；评估图表的质量

3、按照情形，选择输入输出设备

(1) 光学条码阅读器（可将条形码转换成编码输入，合用于商品发卖）

(2) 键盘（直接输入而无兴趣搞前言，简片面，但输入速度较慢）

(3) 磁盘驱动器（逐个或批量输入，成本低、措置效率高、携带便利、速度快）

(4) 光学字符识别（直接阅读合适规范要求的原始凭证上的数字和符号。可识此外有限）

(5) 语音识别输入（使用语音的体例输入数据，但今朝识别率不高，未来输入成长趋向）

(6) 触摸屏（经由过程屏幕的触摸输入信息，便利但输入速度慢）

(7) 光笔输入（将笔画转换成数据或字符）

(8) 鼠标选择输入（适合选择性的场所）

4、几种常见人机对话的选择

(1) 菜单式（操作简单，用户记忆少，但被选项多的话搜索不轻易）；

(2) 呼吁式（用户要熟悉系统才能使用）；

(3) 应答式（对于需要同用户进行交互的）；

(4) 填暗示（易使用、易把握，一般只能用于需要输入数据的场所）

(5) 停笔型（一般仅限于鼠标操作，但占有屏幕的空间较大，表告竣象概念描述力差）

例题：某公司开办了在线电子商务网站，首要为各注册的商家供给在线商品发卖功能。

为更好地吸引用户，该公司打算为注册的商家供给商品促销功能。商品的分类分歧，促销的体例和内容会有所分歧。注册商家可发布促销信息。商家首先要在自己所发卖的商品的分类中，选择促销涉及的某一具体分类，然后选出该分类的一个或多个商品（一种商品仅仅属于一种分类），接着拟定出一个斗劲优惠的折扣政策和促销勾当的优惠时刻，最后由系统生成促销信息并将该促销信息发布在网站上。商家发布促销信息后，网站的注册用户便可经由过程网站采办促销商品。用户可选择介入某一个促销勾当，并选择具体的促销商品，输入采办数目等采办信息。系统生成响应的一份促销订单。只要用户在优惠勾当的时刻规模内，经由

过程网站供给的在线支出系统， 确认在线支出该促销订单 （即完成支出 ），就可以优惠的价钱完成商品的采办勾当，否则该促销订单失踪效。

11. 计算机系统综合知识

1、 计算机组成与体系结构

- 各种计算机体系结构的特点与应用 （SMPMP等）
- 构成计算机的各类部件的功能及其相互关系

2、 操作系统

- 操作系统的类型与结构
- 操作系统基本原理
- 操作系统性能优化
- 网络操作系统与嵌入式操作系统

3、 数据通信与计算机网络

- 数据通信的基本知识
- 开放系统互连参考模型
- 常用的协议标准
- 网络的互连与常用网络设备
- 计算机网络的分类与应用

4、 数据库系统

- 数据库管理系统的类型结构和性能评价
- 常用的关系型数据库管理系统
- 数据仓库与数据挖掘技术
- 数据库工程

5 、 中间件

- 异构与通用服务远程过程调用 （RemoteProcedureCall） 面向消息的中间件 （Message OrientedMiddleware） 对象请求代理 （ObjectRequestBrokers）

6 、 系统配置与性能评价

- C/S 与 B/S 结构分布式系统
- 系统配置方法 （ 双份双重热备份容错集群 ）
- 性能计算 （ 响应时间吞吐量 TAT）

- 性能设计（系统调整 Amdahl 解决方案响应特性负载均衡 ）
- 性能指标（SPECIntSPECFpTPCGibsonmix响应时间 ）
- 性能评估

7、计算机应用

- 信息管理数据处理辅助设计自动控制科学计算
- 远程通信服务 Web计算
- 多媒体压缩编码与存储技术
- 人工智能模式识别

12. 信息化基础知识

1、信息化

- 信息与信息化
- 信息化对组织的意义
- 现代组织对信息化的需求
- 组织的信息化与软件工程或系统集成工程的不同特点

2、政府信息化与电子政务

- 电子政务的概念内容和技术形式
- 中国政府信息化的策略和历程
- 电子政务建设的过程模式和技术模式

3、企业信息化与电子商务

- 企业信息化的概念目的规划方法
- ERP的主要模块和主要算法
- 企业业务流程重组（BPR）
- CRMPD在企业的应用
- 知识管理
- 企业门户
- 企业应用集成外语学习网
- 全程供应链管理的思想
- 商业智能
- 电子商务的类型标准

4、信息资源管理

5、国际和国内有关信息化的标准法律和规定

13. 软件工程

1、软件工程基础知识

- 软件生存周期及其模型
- 软件需求分析与定义
- 软件设计
- 软件测试与审计
- 软件维护
- 软件复用
- 文档编制标准

2、项目管理知识

- 项目计划的制订监督控制
- 项目工作量估算
- 范围管理
- 进度管理
- 配置管理
- 风险管理
- 资源和任务分配
- 项目的生命周期管理

3、软件过程

- 软件过程的定义和范围
- 软件过程的作用
- 软件过程的发展
- 主要的软件过程及其特点
- 软件过程改进

4、质量管理

- 质量保证计划
- 质量认证体系

- 质量管理和质量管理技术
- 全程质量管理

14. 信息系统开发与运行

1、 信息系统基础知识

- 信息系统概念
- 信息系统的功能类型结构
- 信息系统的生命周期各阶段的目标和主要工作内容
- 信息系统建设的原则
- 信息系统开发方法

2、 需求分析和设计方法

- 分析设计图示 (DFD/ERD/流程图 NS图 PAD)
- 面向对象设计 (继承抽象代理封装多态)
- 结构化分析设计
- 模块设计 (内聚耦合)
- I/O 设计 (报表设计屏幕设计数据项代码设计) 人机界面设计

3、 开发环境与开发工具

- 集成开发环境
- 开发工具 (分析设计工具编程工具测试工具等)
- 软件开发平台的比较

4、 系统集成

- 控制集成数据集成表示集成外部资源的使用

5、 应用系统构建

- 应用系统设计开发 (分析设计方法的使用外部设计内部设计程序设计测试)
- 软件包的使用 (开发工具运行管理工具业务处理工具 ERP群件 OA工具)
- 数据库设计和操作 (范式 SQL数据分布)
- 网络的规划与设计

6、 系统运行

- 系统运行管理 (计算机系统数据库网络)
- 系统成本管理 (用户收费 TCO)

	现值计算 货币时间价值			冷备份、热备份	
2009 下半年	软件架构设计 软件质量属性	结构化软件 系统建模 流程图和数据 流程图含义、 区别 高质量数据 流程图三原则	嵌入式软件 体系架构	软件系统架构 架构风格	信息系统安全性 安全威胁 安全认证 授权侵犯
2010 上半年	需求分析 鱼骨图 需求定义文档 内容、作用	宏观经济数 据库建设 原型分析 集中数据库 分布式数据 库	嵌入式软件 可信计算	数据库集成 数据仓库 联邦数据库 Web内容提取、 Web数据挖掘	Web应用系统负载均衡 传输层负载均衡 应用层负载均衡
2010 下半年	软件系统架构 软件架构风格	软件系统数 据架构建模 集中式数据 架构 分布式数据 架构 数据架构扩 展性	基础软件架 构 开放式软件 架构	系统设计与开 发工具集成 ESB基础架构 架构风格 设计模式	信息系统可靠性 可靠度、失效率 动态冗余、N版本程序设 计 常用检错技术
2011 上半年	数字视频监控 告警系统 分层架构 系统安全性、 实时性、稳定 性、扩展性	分布式存储 系统设计 GFS、HDFS 单点失效问 题解决办法	机载信息处 理系统数据 库管理软件 需求变更管 理 数据完整性 保护机制	数据库架构设 计 关系数据库、 NoSQL数据库 NoSQL数据存 储类型 NoSQL数据库 常见问题	网上交易系统 客户端开发、服务端开 发 Ajax 技术 服务端 JavaScript 优势
2011 下半年	软件架构评估 质量属性效用 树 架构风险点、 敏感点、权衡 点	软件系统数 据建模 反规范化技 术定义、优 势、常见问题	嵌入式系统 设计	Web应用系统 架构设计 数据源特征、 数据操作方式 表现层状态转 换 REST	信息系统安全性 对称加密、公钥加密 机密性、完整性保障
2012 上半年	软件项目管理 PERT图、Gantt 图	企业应用系 统开发 服务耦合性 服务重构	嵌入式软件 虚拟化	软件可靠性 故障树方法	Web应用系统开发 .NET 平台、Java 平台 MVC模式 事务控制、EJB 规范

2012 下半年	软件架构风格 软件架构风格 定义 集成开发环境 选取架构风格	分布式系统设计 ISO 开放分布 进程规范 (ODP)	嵌入式软件 架构设计	面向对象系统 建模 设计模式 (创 建型、 结构型、 行为型) 作用	数据架构 NoSQL数据库特点、 常见 问题
2013 上半年	需求建模 用例建模 用例规约 用例关系	桌面云项目 建设 虚拟化技术	嵌入式系统 架构风格	数据库分析与 设计 关系模式 规范化设计 逆规范化方法 水平分割	软件维护 软件维护内容、 特点
2013 下半年	企业应用系统 集成架构设计 服务建模 ESB定义、 5 个 主要功能	项目开发计 划 项目进度保 障措施	嵌入式软件 EMEA方法	表现层设计 MVC模式 设计模式	软件与信息安全 口令认证、 公钥认证 数字信封加密、 文件数 据加密、 块加密、 流加 密 数据库加密解密 API、 透 明加冕
2014 上半年	需求获取方法 联合需求计划 (JRP) 文档分析	项目可行性 分析 项目可行性 分析 4 方面内 容 净现值计算	嵌入式系统 分区化技术	信息化系统维 护 逆向工程 软件重构 3 个 类别、 重构方 法 重构常见问题	Web电子商务系统需求 分析 Flex 和 HTML5 高并发需求常用解决办 法
2014 下半年	Web应用架构 设计 DNS负载均衡、 反向代理负载 均衡 数据库扩展： 数据分区、 主 从复制 数据库缓存、 Memcached分 布式缓存	网络设备管 理系统架构 MVC架构风格	系统过程建 模 数据流图基 本元素、 作 用	嵌入式构件设 计	软件架构评估 质量属性效用树 架构风险点、 敏感点、 权衡点

2004 年-2015 年系统分析师论文题目锦集【完整版】

希赛软考学院 总结的 2004 年-2015 年软考系统分析师考试论文真题题目如下：

2015 年上半年

试题一：论项目风险管理及其应用

试题二：论软件系统测试及其应用

试题三：论软件系统的容灾与恢复

试题四：论非关系型数据库技术及应用

2014 年上半年

试题一：论信息系统开发方法及应用

试题二：论业务流程建模方法及应用

试题三：论数据库集群技术及应用

试题四：论企业信息集成技术及应用

2013 年上半年

试题一：论面向对象建模方法的应用

试题二：论软件企业的软件过程改进

试题三：论企业业务流程优化

试题四：论信息系统的可靠性分析与设计

2012 年上半年

试题一：论软件需求管理及其应用

试题二：论敏捷开发在企业软件开发中的应用

试题三：论信息化建设中的企业知识管理

试题四：论大数据处理技术及其应用

2010 年上半年

试题一：论软件维护及软件可维护性

试题二：论面向服务的企业应用集成技术及其应用

试题三：论快速应用开发在系统建模中的应用

试题四：论信息系统中的访问控制

2009 年上半年

试题一：论软件项目质量管理及其应用

试题二：论企业服务总线技术及其在应用集成中的作用

试题三：论 workflow 管理技术在 CIM 系统协作中的应用

试题四：论政务流程的优化与再造

2008 年下半年

试题一：论基于场景的软件体系结构评估方法

试题二：论敏捷开发方法的应用

试题三：论 SOA 在企业信息化中的应用

试题四：SaaS(Software-as-a-Service , 软件即服务) 的关键技术

2008 年上半年

试题一：论软件体系结构风格及其应用

试题二：论软件项目估算的过程与方法

试题三：论信息资源规划的需求分析

试题四：P2P 计算关键技术与应用

2007 年下半年

试题一：论迭代式软件开发过程与方法

试题二：论 workflow 管理技术在 BPR 中的应用

试题三：论电子商务系统中的技术基础设施集成

试题四：论虚拟计算的应用

2007 年上半年

试题一：论企业软件过程改进的实施

试题二：论信息系统的可行性分析

试题三：论 workflow 管理系统的分析和实现

试题四：动态语言在互联网应用中的作用

2006 年上半年

试题一：论需求获取技术

试题二：论 XML 语言在 Internet 平台上的应用

试题三：论设计模式在软件开发中的应用

试题四：论控制系统的可视化技术

2005 年下半年

试题一：论项目的风险管理

试题二：论 WEB 应用程序的测试

试题三：论电子政务建设中政务内网和外网的划分

试题四：论高可靠性系统中软件容错技术的应用

2005 年上半年

试题一：论企业信息系统的的核心安全

试题二：论电子政务信息共享整合

试题三：论多层分布式结构系统的开发

试题四：论项目管理中的进度控制

2004 年下半年

试题一：论企业内部网的安全策略

试题二：论应用系统开发范围和功能的确定

试题三：论嵌入式操作系统的功能与特性

试题四：论软件测试计划的制定

2004 年上半年

试题一：论电子商务的安全

试题二：论用例的获取方法

试题三：论 ERP 的开发与应用

试题四：论软件开发成本估算

四、案例范围

1.系统计划：项目提出与选择；项目优先级确定；业务评价；分析系统的实施目的、功能、架构、规模、能力、维护、应用方法及故障情况等；系统开发规划；可研；成本效益分析；系统方案制定；评价和改进；新旧系统的分析和比较；计划变更与控制

2.需求工程：

3.系统分析：组织结构分析；功能分析；业务流程分析；数据流程分析；系统功能划分；系统可靠性模型；系统故障模型；可靠性分析与度量；系统故障与恢复对策；

4.建模技术：需求建模步骤；用例驱动的开发方式；结构化建模技术；数据流图；面向对象建模技术；数据库建模

5.系统设计：

6.文档编写：

7.沟通能力：

8.系统运维：

9.软件过程改进：

10.项目管理：

11.企业信息化战略与实施：

五、论文范围

- 1.信息系统工程：系统计划、系统分析、需求工程、系统测试、系统维护、项目管理、质量保证、面向对象设计、软件过程改进、产品线应用与设计
- 2.数据库及应用：数据库管理、分析、建模、应用、数据仓库、数据集市、数据挖掘
- 3.网络规划与应用：
- 4.系统安全
- 5.新技术
- 6.应用系统集成
- 7.企业信息系统：电子商务、电子政务、DSS决策支持系统
- 8.企业信息化的组织和实施

六、9 大管理输入输出

一、 项目整体管理		
1、制订项目章程； 2、制订项目范围说明书（初步） ；3、制订项目管理计划； 4、指导和管理项目实施； 5、监督和控制项目工作； 6 整体变更控制； 7、项目收尾		
1、制订项目章程		
输入	工具与技术	输出
?合同：来自客户的采购组织 ?工作说明书（ SOW）：对内部项目：基于业务需要或产品 （服务）的需求。对外部项目：作为投标文件的一部分从客户那里得到。 ?环境的和组织的因素 ?组织过程资产。	?项目选择方法 收益测量方法（即对比法、评分模型、收益分布或经济模型）； 数据模型； 决策模型、 ?项目管理方法：帮助项目管理团队有效地制定项目章程。 ?项目管理信息系统（ PMIS） ?专家判断	项目章程
2、初步范围说明书		
输入	工具与技术	输出
?项目章程 ?工作说明书（ SOW） ?环境与组织因素 ?组织过程资产	?项目管理方法论。 ?项目管理信息系统（ PMIS）。 ?专家判断	制定出范围说明书
3、制订项目管理计划		
输入	工具与技术	输出
项目范围说明书（初步） 项目管理过程 环境和组织因素 组织过程资产	?项目管理方法论。 ?项目管理信息系统（ PMIS） ?专家判断	?项目管理计划
4、指导和管理项目实施		
输入	工具与技术	输出

1、项目管理计划 2、已批准的纠正措施 3、已批准的预防措施 4、已批准的变更申请 5、已批准的缺陷修复 6、确认缺陷修复 7、管理收尾规程	?项目管理方法论 ? 项目管理信息系统 (PMIS)	?可交付物 ?申请的变更 ?已实施的变更申请 ?已实施的纠正措施 ?已实施的预防行动 ?已实施的缺陷修复 ?工作绩效信息
5、监督和控制项目工作		
输入	工具与技术	输出
?项目管理计划 ?工作绩效信息 ?被拒绝的变更需求	?项目管理方法论 ?项目管理信息系统 (PMIS) ?挣值管理 ?专家判断	?建议的纠正措施 ?建议的预防措施 ?预测 ?建议的缺陷修复 ?变更申请
6、整体变更控制		
输入	工具与技术	输出
?项目管理计划 ?申请的变更 ?工作绩效信息 ?建议的预防措施 ?建议的纠正措施 ?建议的缺陷修复 ?可交付物	?项目管理方法论 ?项目管理信息系统 (PMIS) ?专家判断	?已批准的变更申请 ?被拒绝的变更申请 ?项目管理计划更新 ?项目范围说明书更新 ?已批准的纠正措施 ?已批准的预防措施 ?已批准的缺陷修复 ?可交付物
7、项目收尾		
输入	工具与技术	输出
1、项目管理计划 2、合同文件 3、企业环境因素 4、组织过程资产 5、工作绩效信息 6、可交付物	?项目管理方法论。 ?项目管理信息系统 (PMIS) ?专家判断	?管理收尾规程 ?合同收尾规程 ?最终产品、服务或成果和项目结果 ?组织过程资产 (已更新)

二、 项目范围管理		
1、范围计划； 2、范围定义； 3、工作分解结构； 4、范围确认； 5、范围控制；		
1、范围计划 范围计划是指进一步形成各种文档，为将来项目决策提供基础，这些文档中包括用以衡量一个项目或项目阶段是否已经顺利完成的标准等。作为范围计划过程的输出，项目组要制定一个范围说明书和范围管理计划。		
输入	工具与技术	输出
?企业环境因素 ?组织过程资产； ?项目章程	?专家判断 ?模板、表格和标准	范围管理计划

?项目范围说明书（初步） ?项目管理计划		
2、范围定义：范围定义是指将项目主要的可交付成果细分成较小的、更易管理的组分。范围定义最重要的任务就是详细定义项目范围边界。		
输入	工具与技术	输出
?组织过程资产 ?项目章程 ?项目范围说明书（初步） (4) 项目范围管理计划 (5) 批准的变更申请	?产品分析 ?可选方案识别 ?专家判断法 (4) 项目干系人分析	?项目范围说明书（详细） ?变更请求 (3) 项目管理计划（更新）
3、创建工作分解结构 工作分解结构（WBS）是面向可交付物的项目元素的层次分解，详细描述了项目所要完成的工作。WBS 的最低层次元素是能够被评估的、安排进度的和被跟踪的。它是组织管理工作的主要依据。		
输入	工具与技术	输出
?组织过程资产 ?项目范围说明书 ?项目范围管理计划 (4) 批准的变更申请	1、WBS模板 2、分解技术	1、范围说明书 2、WBS 和WBS 字典 3、范围基线 4、项目管理计划（更新） 5、变更申请
4、范围确认		
输入	工具与技术	输出
?项目范围说明书 项目范围管理计划 ? WBS 和WBS 字典 ?项目范围管理计划 ?可交付物	检查	1、已接受的交付物 2、变更申请 3 推荐的纠正措施
5、范围控制		
输入	工具与技术	输出
1、范围说明书 2、范围管理计划 3、WBS 和WBS 字典 4、绩效报告 5、工作绩效信息 6、批准的变更需求 7、工作绩效信息	?变更控制系统 ?偏差分析 ?重新规划 ?配置管理系统	1、范围说明（新） 2、WBS 和WBS 字典（更新） 3、范围基线更新 4、变更申请 5、建议的纠正措施 6、组织过程资产（更新） 7、项目管理计划（更新）

三、项目时间管理 1、活动定义； 2、活动排序； 3、活动资源估算； 4、活动历时估算； 5、制订进度计划； 6、进度控制；		
1、活动定义 为得到工作分解结构（WBS）中最底层的交付物执行的一系列活动，对这些活动的识别以及归档的过程就做活动的定义		
输入	工具与技术	输出
1、企业环境因素 2、组织的过程资产	1、分解 2、模板	?项目活动清单 ?活动清单属性

3、项目范围说明书 4、WBS WB字典 5、项目管理计划	3、滚动式规划 4、专家判断 5、规划组成部分	?里程碑清单 ?发起的变更
2、活动排序：活动排序也称为工作排序，即确定活动之间的依赖关系，形成文档。		
输入	工具与技术	输出
1、活动清单，即在活动定义过程所得的结果 2、活动清单属性 3、项目范围说明书 4、里程碑清单 5、批准的变更请求	1、前导图法 2、箭线图法 3、进度计划网络模板 4、确定依赖关系 5、提前、滞后	1、项目计划网络图 2、项目活动清单 3、活动清单属性 4、发起的变更
3、活动资源估算		
输入	工具与技术	输出
1、企业环境因素 2、组织的过程资产 3、活动清单 4、活动属性 5、资源可用性 6、项目管理计划	?专家判断法 ?替换方案确定 ?公开的估算数据 ?估算软件 ?自下而上的估算	1、活动资源需求 2、更新过的活动属性 3、资源分解结构（RBS） 4、更新的资源日历 5、变更请求
4、活动历时估算。		
输入	工具与技术	输出
1、企业环境因素； 2、组织的过程资产； 3、项目范围说明书； 4、活动清单； 5、 活动清单属性； 6、活动资源需求； 7、 资源日历； 8、项目管理计划；	1、专家判断 2、类比估算法 3、参数式估计 4、历时的三点估算 5、预留时间	?活动历时估算结果 ?活动清单、属性（更新）
5、制订进度计划		
输入	工具与技术	输出
1、组织的过程资产； 2、项目范围说明书； 3、活动清单、活动属性； 4、项目进度网络图； 5、活动资源要求； 6、资源日历； 7、活动历时估算； 8、项目管理计划	1、进度网络分析； 2、关键路径法； 3、进度压缩； 4、假设情景分析； 5、资源平衡； 6、关键链； 7、项目管理软件； 8、所采用的日历； 9、超前和滞后； 10、速度模型；	1、项目进度计划（进度网络图、甘特图、里程碑图）；2、速度模型数据； 3、进度基准； 4、资源需求更新； 5、活动属性更新； 6、项目日历更新； 7、请求的变更； 8、项目管理计划更新；
6、进度控制		
输入	工具与技术	输出
1、项目进度计划 2、进度基准 3、绩效报告 4、已批准的变更需求	1、进展报告； 2、进度更新控制系统； 3、绩效测量； 4、项目管理软件； 5、偏差分析； 6、计划比较甘特图；	1、速度模型数据更新； 2、进度基准更新； 3、绩效衡量； 4、变更需求； 5、建议的纠正措施； 6、组织过程资产； 7、活动清单、属性； 7、项目管理计划；

四、 项目成本管理		
1、成本估算； 2、成本预算； 3、成本控制；		
1、成本估算		
输入	工具与技术	输出
1、企业环境因素； 2、组织过程资产； 3、项目范围说明书； 4、WBS WBS字典； 5、项目管理计划（进度管理计划、员工管理计划、风险事件）；	1、类比估算法（自上而下法）；2、确定资源费率； 3、自下而上成本估算（工料清单法）；4、项目管理软件；5、卖方投标分析； 6、准备金分析； 7、质量成本；	1、项目成本估算结果 2、相关支持性细节文件和结果 3、请求的变更 4、成本管理计划更新
2、成本预算		
输入	工具与技术	输出
1、目范围说明书； 2、WBS WBS字典； 3、活动成本估算； 4、动成本估算的支持性细节； 5、项目进度计划；	成本总计 管理储备 参数模型 支出的合理法原则	1、成本基准计划 2、项目资金需求 3、项目管理计划（更新） 4、请求的变更
3、成本控制		
输入	工具与技术	输出
1、成本基准；、 2、项目资金需求； 3、成本绩效报告； 4、工作绩效报告； 5、批准的变更申请； 6、项目管理计划	1、成本变更控制系统； 2、绩效测量； 3、预测技术； 4、项目绩效评估（偏差、趋势、挣值分析）； 5、项目管理软件； 6、偏差管理；	1、成本估算； 2、成本基线更新； 3、预测完工； 4、绩效衡量； 5、请求的变更； 6、建议纠正措施； 7、组织过程资产（新）；8、项目管理计划（新）

四个关键值

PV(计划值)：成本估算部分的总价值；

AC(实际成本)：在规定时间内，完成的成本总额；

EV(挣值)实际完成工作的预算价值，即到某一点已完成工作应当的投入资金。

ETC(剩余工作的成本估算) $ETC = \text{总的 PV} - \text{已完成的 EV}$ ； $ETC = \text{剩余工作的 PV} \times CPI$
最常用的尺度：

CV(成本偏差)： $CV = EV - AC$ $CV > 0$ (成本节省) $CV < 0$ (成本超支)

SV(进度偏差)： $SV = EV - PV$ $SV > 0$ (超过进度) $SV < 0$ (落后进度)

CPI(成本绩效指数)： $CPI = EV / AC$ $CPI > 1.0$ (成本节余) $CPI < 1.0$ (成本超支)

SPI(进度绩效指数)： $SPI = EV / PV$ $SPI > 1.0$ (进度超前) $SPI < 1.0$ (进度滞后)

五、 项目质量管理		
1、质量规划； 2、质量保证； 3、质量控制；		
1、质量规划		
输入	工具与技术	输出
项目章程	成本 /效益分析	质量管理计划
项目管理计划	基准分析：	质量度量指标

项目范围说明书 （用来定义项目干系人需求，阈值和接受标准） 组织过程资产 环境何组织因素	实验设计 质量成本	质量检查单 过程改进计划 项目管理计划（更新）
2、质量保证。		
输入	工具与技术	输出
质量管理计划 质量度量标准 （清晰的规划说明和完善的标准） 过程改进计划 工作绩效信息 变更请求 质量控制测量	质量计划工具和技术 （成本 /效益分析、基准分析、实验设计、 质量成本） 质量审计 过程分析 质量控制工具和技术（检查、控制图、帕累托图、统计抽样、流程图、趋势分析） 基准分析	请求的变更 建议的纠正措施 组织过程资产 （更新） 项目管理计划 （更新）
3、质量控制。		
输入	工具与技术	输出
质量管理计划 质度量标准 质量检查表 组织过程资产 工作绩效信息 已批准的变更请求 产品、服务和结果	1、检查； 2、控制图； 3、帕累托图； 4、抽样统计； 5、流程图（鱼刺图）； 6、趋势分析； 7、缺陷修复审查；	1、建议的纠正措施 2、建议的预防措施 3、请求的变更 4、建议的缺陷修复 5、已确认的缺陷修复 6、项目管理计划（更新） 7、质量控制质度量 8、组织过程资产（更新）

预防（把错误排除在过程之外）和检查（把错误排除在到达客户之前）

属性抽样（结果合格或不合格）和变量抽样（按量度合格度的连续尺度衡量所得结果）。

特殊原因（异常事件）和随机原因（正常过程偏差）。

许可误差（如果其结果落入误差范围所界定的范围内，那么这个结果就是可接受的）和控制界限（如果其成果落入控制界限内。那么该项目在控制之中）。

六、 人力资源管理		
1、人力资源计划编制； 2、组建项目团队； 3、项目团队建设； 4、管理项目团队		
1、人力资源计划编制		
输入	工具与技术	输出
活动资源估计 环境和组织因素 项目管理计划	组织结构图和职位描述 人力资源模板 人际网络 组织理论：	角色和职责 项目的组织结构图 人员配备管理计划：
2、组建项目团队；		
输入	工具与技术	输出
角色和职责 项目的组织结构图 人员配备管理计划 环境的和组织因素	事先分派 谈判 采购 虚拟团队	项目人员分配 资源日历 人员配备管理计划（已更新） 资源可用性

组织过程资产		
3、项目团队建设；		
输入	工具与技术	输出
项目人员分配 人员配备管理计划	一般管理技能 培训 团队建设活动 基本原则 同地办公 （集中） 认可和奖励	团队绩效评估 （提高个人技能；提高团队能力；较低的员工流动率）
4、管理项目团队		
输入	工具与技术	输出
项目人员分配 角色和职责 项目的组织结构图 人员配备管理计划 绩效报告 组织过程资产	观察和对话 项目绩效评估 冲突管理（问题解决、妥协、求同存异、撤退、强迫） 问题日志	人员配备管理计划（更新） 变更请求 组织过程资产（更新）

七、项目沟通管理		
1、沟通计划编制； 2、信息分发； 3、绩效报告； 4、项目干系人管理		
1、沟通计划编制		
输入	工具与技术	输出
组织过程资产 项目章程 项目管理计划 项目范围说明书）	项目干系人分析 沟通需求分析 沟通技术	1、 沟通管理计划
2、信息分发。		
输入	工具与技术	输出
工作绩效信息 沟通管理计划	1、沟通技术（正式沟通、非正式沟通） 2、信息收集和检索系统 3、信息分发方法 4、取得的经验教训	更新的项目管理计划 组织过程资产
3、绩效报告（状态、进展报告） 、		
输入	工具与技术	输出
工作绩效信息 项目管理计划 预测 已批准的变更请求 可交付物	信息表示工具 绩效信息收集和编辑 状态评审会议	绩效报告 预测 需求变更 更新的项目管理计划 建议的纠正措施 组织过程资产
4、项目干系人管理		
输入	工具与技术	输出
项目管理计划 沟通管理计划	1、沟通方法 2、问题日志	问题解决 更新的项目沟通管理计划

组织过程资产		组织过程资产
--------	--	--------

团队内沟通

1、事前准备 2、确认需求 3、阐述观点 4、处理异议 5、达成协议 6、共同实施

把握项目沟通基本原则

- 1、沟通内外有别 2、非正式沟通有利于关系融洽 3、采用对方能接受的沟通风格 4、沟通升级原则（和对方沟通；和对方上级沟通；和自己上级沟通；自己的上级和对方上级沟通）

冲突的化解与处理（查清冲突的具体原因、公平原则、选择处理的策略、尽量采用双赢原则）

冲突的管理与防范（1、亲自解决争端；2、鼓励坦率的情感表达；3、确立准则、职务示范和劝导；4、利用冲突的意识；5、抑制和控制潜在冲突）

八、项目风险管理		
1、风险管理计划编制； 2、风险识别； 3、定性风险分析； 4、定量风险分析； 5、风险应对计划编制； 6、风险监控与跟踪		
1、风险管理计划编制		
输入	工具与技术	输出
项目章程 项目范围说明书 项目管理计划 组织过程资产 环境和组织因素	1、计划会	方法论； 角色和职责； 预算 制订时间表； 风险类别； 风险概率和影响力的定义； 概率及影响矩阵 已修订的项目干系人对风险的容忍度； 报告的格式； 跟踪
2、风险识别		
输入	工具与技术	输出
项目章程； 项目范围说明书； 项目管理计划； 组织过程资产； 环境及组织因素	1、文档评审； 2、信息收集技术； 3、检查表； 4、假设分析； 5、 图解技术；	1、风险记录； 2、项目管理计划
3、定性风险分析		
输入	工具与技术	输出
项目管理计划 (包括风险管理计划、风险记录) 组织过程资产； 工作绩效信息 项目范围说明	风险概率及影响评估； 概率 - 影响矩阵 风险数据质量评估； 风险种类 风险紧急度评估	1、风险记录（新）
4、定量风险分析		
输入	工具与技术	输出
项目管理计划 组织过程资产 风险记录	1、数据收集和表示技术 访谈 概率分布 专家判断 2、定量风险分析和建模技术 灵敏度分析 期望货币价值分析 (EMV) 决策树分析 建模和仿真	更新的风险记录 项目可能性分析 实现成本和进度目标的可能性 已量化风险优先级列表 定量风险分析结果中的趋势
5、风险应对计划编制		
输入	工具与技术	输出

风险管理计划； 风险记录	1、负面风险（威胁）的应对策略（避免、转移、减轻） 2、正面风险（机会）的应对策略（开拓、分享、强大） 3、同时适用威胁和机会的应对策略 4、应急响应策略	1、 风险记录（新） 2、 风险相关合同协议
6、风险监控、跟踪 在项目的整个生命期内，监视残余风险，识别新的风险，执行降低风险计划，以及评价这些工作的有效性。		
输入	工具与技术	输出
1、项目管理计划 2、工作绩效信息 3、批准的变更请求	1、风险评估 2、风险审计与定期的风险评估； 3、差异和趋势分析； 4、技术的绩效评估； 5、预留管理	1、纠正措施； 2、变更申请； 3、风险记录（新）； 4、组织过程资产；

九、 项目采购管理 1、采购计划编制； 2、编制合同； 3、招标； 4、供方选择； 5、合同管理； 6、合同收尾		
1、采购计划编制		
输入	工具与技术	输出
项目章程； 项目范围说明书； 项目管理计划； 工作分解结构和字典； 环境因素和组织因素； 组织过程资产； 风险记录	自制 /外购分析； 专家判断 合同类型	采购管理计划； 工作说明书（SOW） 自制 / 外购决定； 项目管理计划（新）
2、编制合同		
输入	工具与技术	输出
采购管理计划； 工作说明书（SOW） 项目管理计划； 自制 /外购决定	标准表格 专家判断	1、采购文档； 2、评估标准； 3、工作说明书（新）
3、招标		
输入	工具与技术	输出
1、 采购文档； 2、组织过程资产	1、投标人会议； 2、广告	1、合同卖方清单； 2、采购文件包； 3、建议书
4、供方选择		
输入	工具与技术	输出
建议书； 评估标准； 组织过程资产 风险数据库； 风险相关的合同协议 合格卖方清单； 采购文件包	1、加权系统； 2、独立核算 3、筛选系统； 4、合同谈判	1、选择的供方； 2、合同 3、合同管理计划； 4、资源可用性；
5、合同管理		
输入	工具与技术	输出
合同 绩效报告 已批准的变更申请 工作绩效信息	合同变更控制系统 买方主持的绩效评审 检查和审计 绩效报告	组织过程资产（更新） 信函；支付进度和请求； 卖方绩效评估文件； 请求的变更 建议的纠正措施

选择供方	支付系统 索赔管理 记录管理系统	合同文件
6、合同收尾		
输入	工具与技术	输出
合同文件 合同收尾过程	过程审计 记录管理系统	合同文件（管理收尾） 正式验收和收尾 经验教训