调制解调：模拟信号转数字信号，再将数字信号转模拟信号

CMMI(能力成熟度模型):

初始级：过程不可预测

已管理级：项目级管理

已定义级：组织级管理，标准过程化，文档化

量化管理级：数据驱动决策，过程可预测

优化级：持续改进，创新导向

## 服务架构

2层C/S架构(表现层，数据层)：前端直连数据库

3层C/S架构(表现层，应用服务，数据层)：业务逻辑提取到服务器中(如java)

架构风格

管道过滤器：

层次结构：降低性能

事件驱动：

黑板风格：公共区域(黑板)交换信息，获得答案

## 管理架构

ERP资源流：物流，资金流，信息流

## 计算机网络

无干扰下最大传输速率(奈奎斯特公式)=2\*带宽B\*log2(信号状态数V) bps

有干扰下最大传输速率(香农公式)=带宽\*log2(1+S/N) bps

## 计算机组成原理

存储器

ROM：出场自带，不可编程

PROM：可编程一次

EPROM： 可多测编程

EEPROM：可多次编程

## 操作系统

## 软件工程

内聚类型(低到高)

偶然：全放一块

逻辑：逻辑内聚(根据不同入参，执行增删改)

时间：相同时刻的步骤放一块(init中操作)

过程：几个无关的过程放在一个方法中(数据无关)

通信：几个无关的过程放在一个方法中(数据有关)

顺序：多个有先后顺序的步骤放一个方法中

功能：包括步骤涉及的其他数据联动操作

耦合类型

无：通过mq,中间件通信，不指定对方存在

数据：模块通过参数传递数据

标记：模块通过参数传整个对象，但只用个别字段

控制：模块A向B传递flag标志，B根据flag执行逻辑

外部：模块依赖外部系统(不同环境结果不同)

公共：多个模块共享全局变量

内容：模块A之间修改B中的数据

4+1视图

进程视图：多进程怎么交互(性能)

物理/部署视图：服务器部署

逻辑视图：功能怎么分

实现/开发视图：代码怎么实现

场景视图：活动图，序列图，使用场景

**开发模型**

瀑布模型：软件需求明确

原型(演化)模型:需求不明确，规模不太大

螺旋模型:原型模型(基础)+瀑布模型，是快速原型模型的改进，分为目标设定、风险分析、开发和有效性验证、评审

增量模型：尽早得到用户反馈

构件模型:重用提高软件可靠性和易维护性

喷泉模型：用户需求为动力，面向对象开发

快速应用开发：系统模块化程度较高时使用

形式化方法：基于数学

敏捷开发:是适应性的，以人为中心，原型模型为基础，增量式开发

极限编程：交流，朴素，反馈，勇气

水晶法：以人为中心

并列争球法：增量迭代，按需求优先级实现

特征驱动：人，过程，技术

净室软件：基于函数和抽样理论，开发前解决缺陷，一次成功

需求跟踪

正向跟踪:从需求上找根因

逆向跟踪：从代码推理需求

用例模型：

包含关系：A包含B的行为

扩展关系：A比B在行为上有扩展(异常时)

泛化关系：A是B的特例(子类)

## 测试

AB测试：随机分组访问不同版本

基准性能测试：

压力测试：能有多少并发

可靠性测试：能无故障跑多久

MTTF: 平均两次故障间运行的时间

MTTR:平均修复时间

MTBF:平均故障间隔(MTTF+MTTR)

## 法律

国家机密保密期：20年

## 数据库

模式

内模式(机器): 分区，页，块，底层结构

模式/概念模式(开发)：索引，约束,表结构

外模式(用户)：视图，数据





