软件与硬件的区别：本质逻辑与物理；软件是设计开发的；软件不会磨损；大部分软件是按需定制的。

IEEE定义：(1)将系统化、规范化、可量化的方法应用于软件的开发、运行和维护，即将工程化方法应用于软件；(2) 在(1)中所述方法的研究。

框架活动:沟通、策划、建模、构建、部署

成熟级别：第0级：不完全级、1已执行级、2已管理级、3已定义级、4已定量管理级、5优化级

软件生命周期：软件计划与可行性研究、需求分析、软件设计、编码、软件测试、运行与维护

瀑布模型：特点—文档驱动 优点：消除非结构化软件；降低软件的复杂度，促进软件开发工程化

缺点：实际项目开发中很少遵守瀑布模型提出的顺序；客户难以清楚的描述真正的需求；客户要等到开发周期的晚期才能看到程序运行的测试版本 ；在线性过程的开始和结束，容易发生“阻塞状态”

RAD缺点：1、对于大型项目，需要大量人力资源来创建相对独立的RAD团队

2、如果开发者和客户没有做好短时间急速完成系统的准备，则可能导致失败

3、因为是构件式开发，如果一个系统不能合理的模块化，会带来很多问题

4、如果系统需求是高性能的，并且需要通过调整构件接口的方式来提高性能，则不能采用RAD模型

5、技术风险高的情况下，不宜采用RAD模型，如项目开发使用大量的新技术

敏捷团队成员特点：基本能力、共同目标、精诚合作、决策能力、模糊问题解决能力、

相互信任和尊重、自我组织

4个框架活动：策划、设计、编码和测试 设计原则：KIS

结对编程：两位程序员肩并肩地坐在同一台电脑前合作完成同一个设计

优点：结对的两人完成其工作，所开发代码和其他工作集成。有些情况下，这种集成工作由集成团队按日实施，还有一些情况下，结对者自己负责集成，这种“连续集成”策略有助于避免兼容性和借口问题，建立能及早发现错误的“冒烟测试”环境

Scrum原则：1组织小团队，以达到“沟通最大化，负担最小化，非语言描述、非形式化知识”

2过程对技术和业务变化具有适应性，以“保证制造具有最好可能的产品”

3过程生产频繁发布“可检查、可调整、可测试、可文档化、可构建”的软件增量

4开发工作和开发人员划分为“清晰的、低耦合的部分或包”

5坚持在产品构建中进行测试和文档化

6提供在任何情况下都能完成产品的能力

宏要素：基于计算机的系统，它作为更大的基于计算机的系统的一部分

系统工程的层次结构：全局视图、领域视图、要素视图、详细视图

导出需求遇到的问题：范围问题、理解问题、易变问题

协同需求收集会议的基本原则：1软件工程师和客户共同举办和参与 2制定筹备与参与会议的规则

3拟定一个会议议程：既涵盖重点，又鼓励自由交流 4由一个主持人控制会议

5使用某种“定义机制”：工作表、活动挂图、不干胶贴纸，电子公告牌、聊天室、虚拟论坛

分析模型的三个目标：1）描述客户需要什么2）为软件设计奠定基础

3）定义在软件完成后可以被确认的一组需求

实体类： 表示系统存储和管理的永久信息；描述必须存贮的信息及其相关行为；通常对应现实世界中的“事物”

边界类 ： 表示参与者与系统之间的交互；描述外部的参与者与系统之间的交互

控制类 ： 表示系统在运行过程中的业务控制逻辑

设计质量的指导原则：

1 设计应展示出这样一种结构：a已经使用可使别的系统风格或模式创建b由展示出良好设计特征的构件构成c能够以演化的形式实现，从而便于实现和测试

2设计应该模块化 3设计应该包括数据、体系结构、接口和构件的清楚的表示

4设计应导出数据结构，这些数据结构适于要实现的类，并由可识别的数据模式提取

5设计应导出显示独立功能特征的构件

6设计应导出接口，这些接口降低了构件之间以及与外部环境连接的复杂性

7设计的导出应根据软件需求分析过程中获取的信息采用可重复使用的方法进行

8应使用有效传达其意义的表示法来表达设计

功能独立：两个标准：内聚和耦合。内聚性显示了某个模块相关功能的强度，耦合性显示了模块的

相互依赖性

数据流类型决定映射方法：变换映射、事务映射

基于类的构件设计基本原则：开关、Liskov替换、依赖倒置、接口分离、发布复用等价性、

共同封装、共同复用原则。

功能内聚：主要通过操作来表现，当一个模块完成一种且只一种运算并返回结果时发生这个级别上的内聚

数据耦合：当操作需要传递较长的数据参数时就会发生

设计概念（重要）

抽象 过程抽象：指具有明确和有限功能的指令序列 数据抽象：描述数据对象的冠名数据集合

体系结构（扇入、扇出、宽度、深度：腰鼓形）模式（设计模式）模块化**：**分而治之的策略（高内聚低耦合）

信息隐蔽 原则**：**每个模块都对其他模块隐藏自己的设计决策

求精自顶向下地设计策略重构重新组织的技术，简化构建的设计，而无需改变其功能或行为

设计类：分类（5个）和特征（4个）

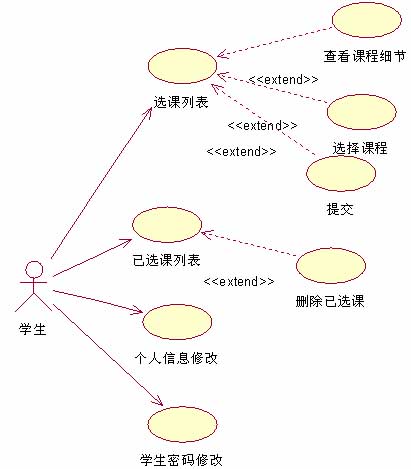
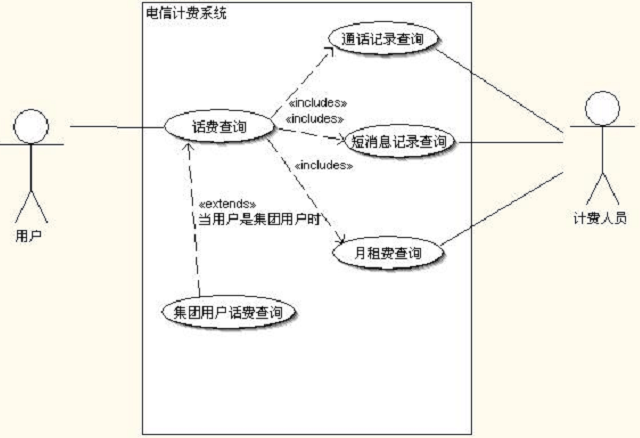
用户接口类 业务域类 过程类 持久类 系统类

完整性与充分性 原始性 高内聚性 低耦合性

黑盒测试 在软件接口处执行测试，检查系统的基本方面，很少关心软件的内部结构

白盒测试 基于过程细节的封闭检查，提供检查特定条件集和（或）循环的测试用例，测试贯穿软件的逻辑路径和构件间的协作。侧重于内部逻辑的测试 语句覆盖 路径覆盖

用例图



**电信计费用例图 学生选课系统用例图**

数据流图



类图 电梯的分类组成、交通工具概念体系、计算机系统组成



状态图电水壶、

计算机、打印机、

复印机的工作



**异常1 异常2**

顺序图 饮料自动销售系统、ATM机取款等等



**正常顺序**