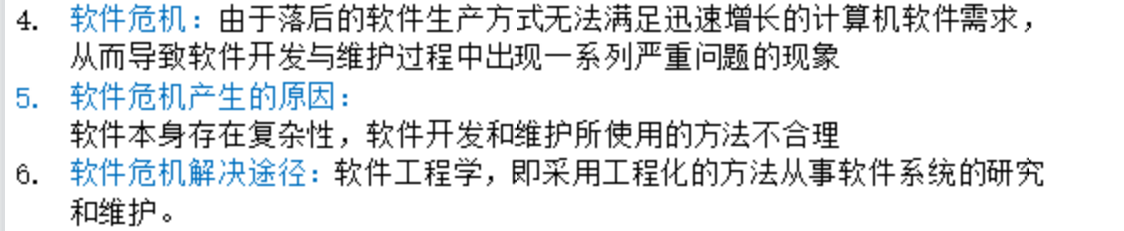
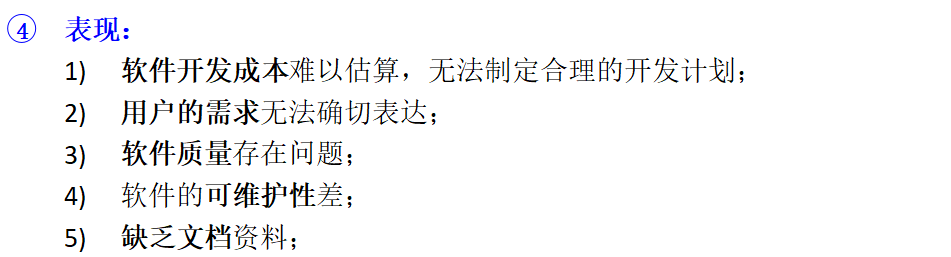
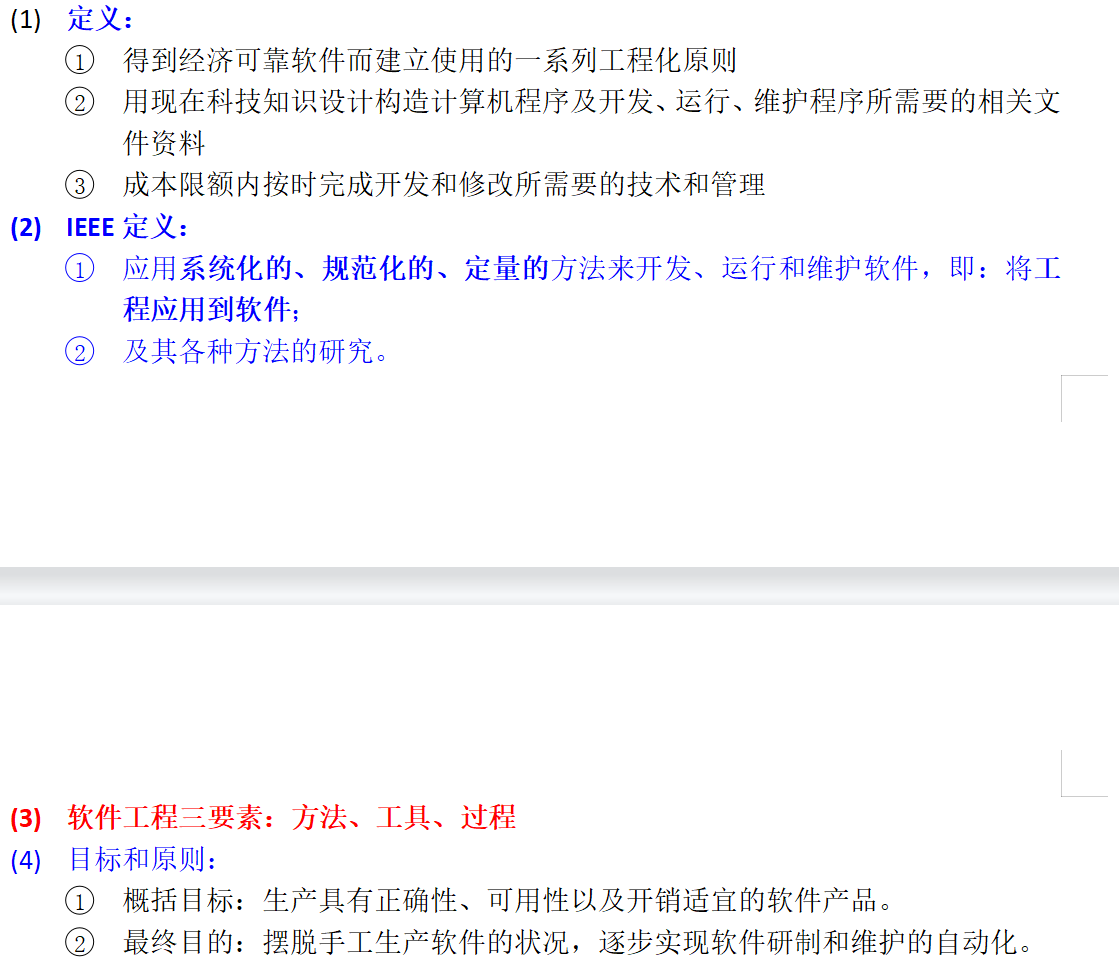
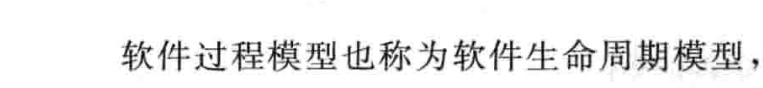
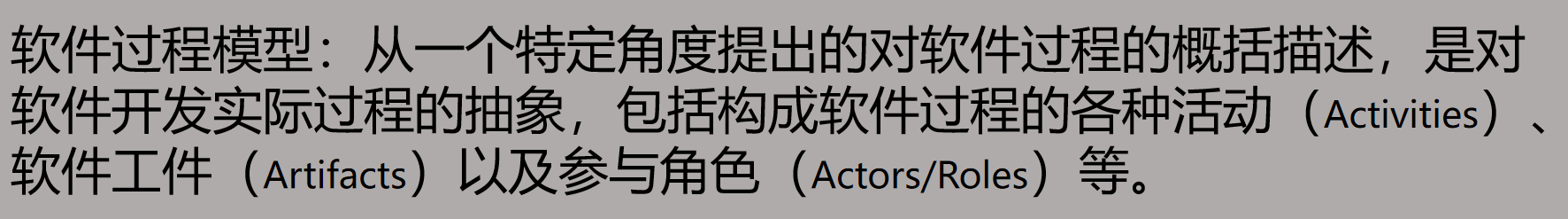
**1.软件工程的起源：软件危机**

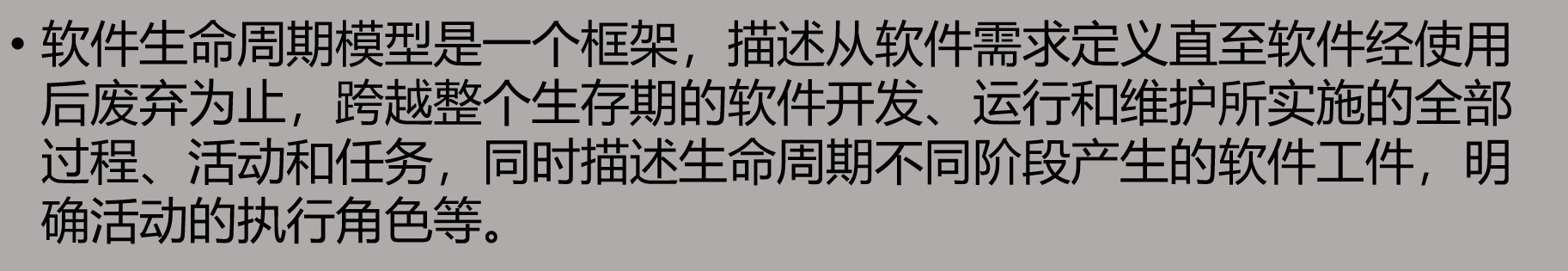


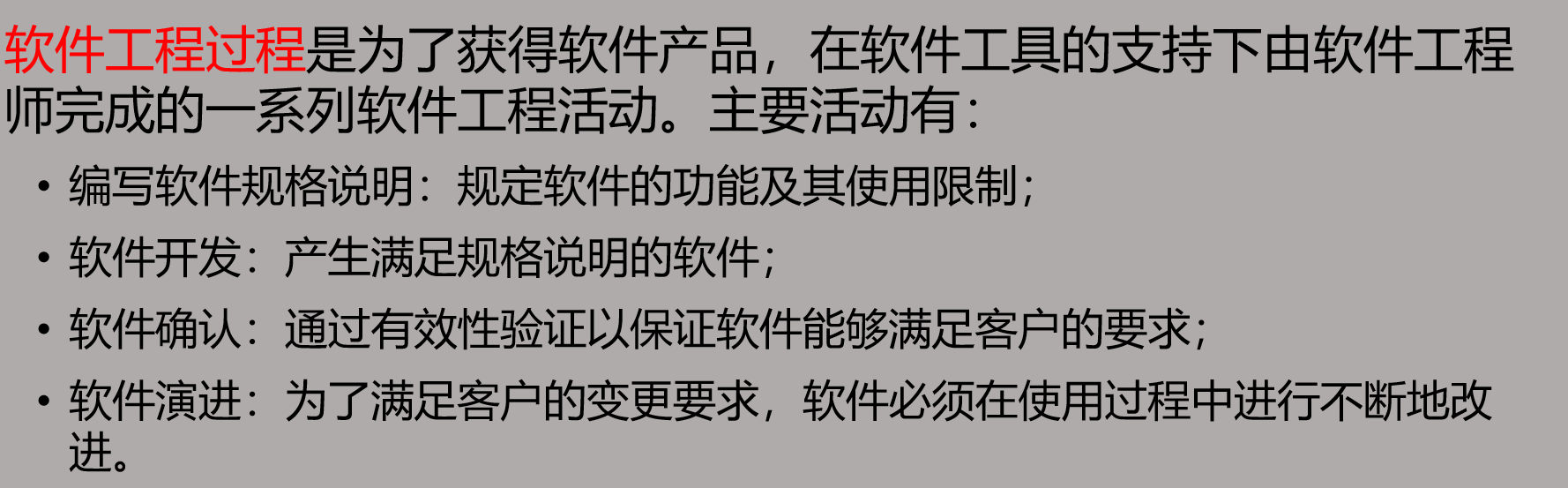
1. **软件工程的定义，任务和目标：**  
   

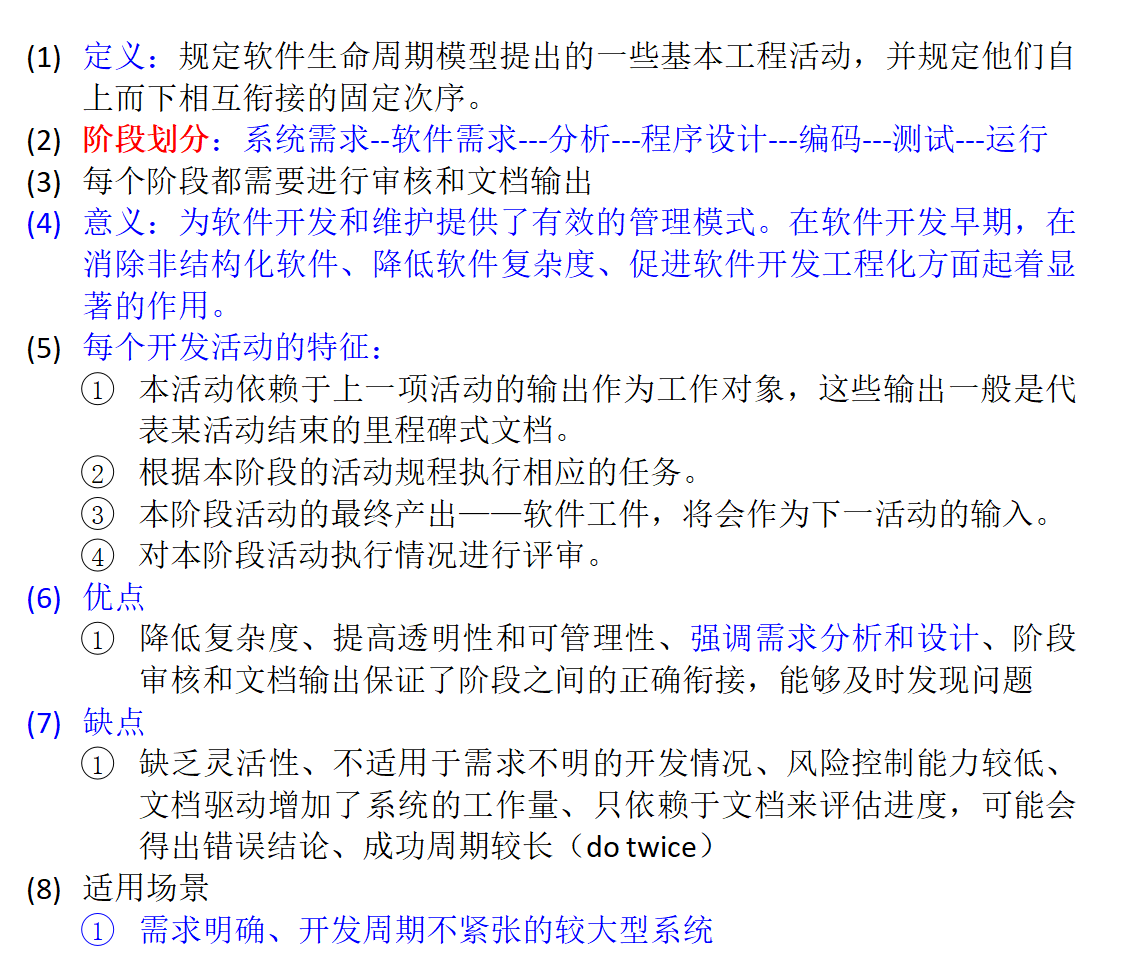
**3.软件生命周期模型的定义，作用和基本活动：**





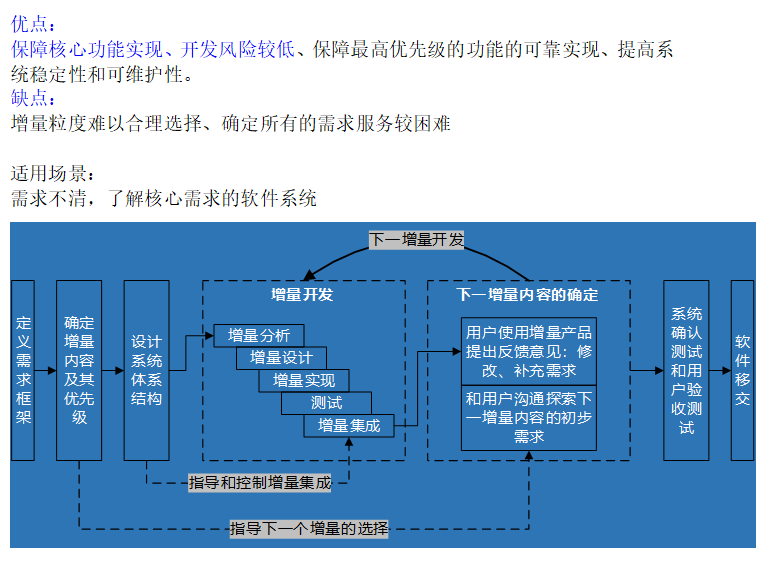
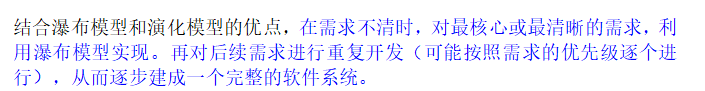


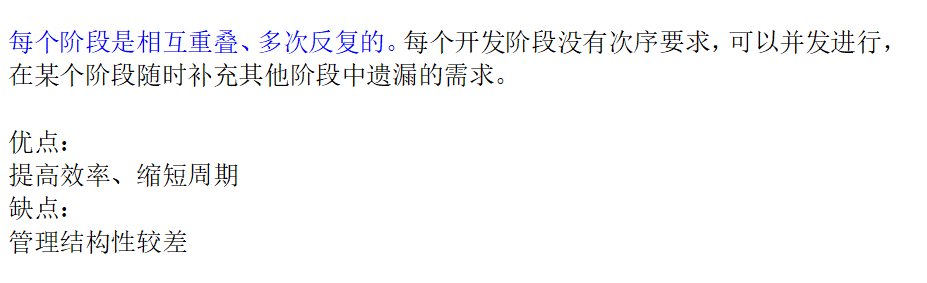


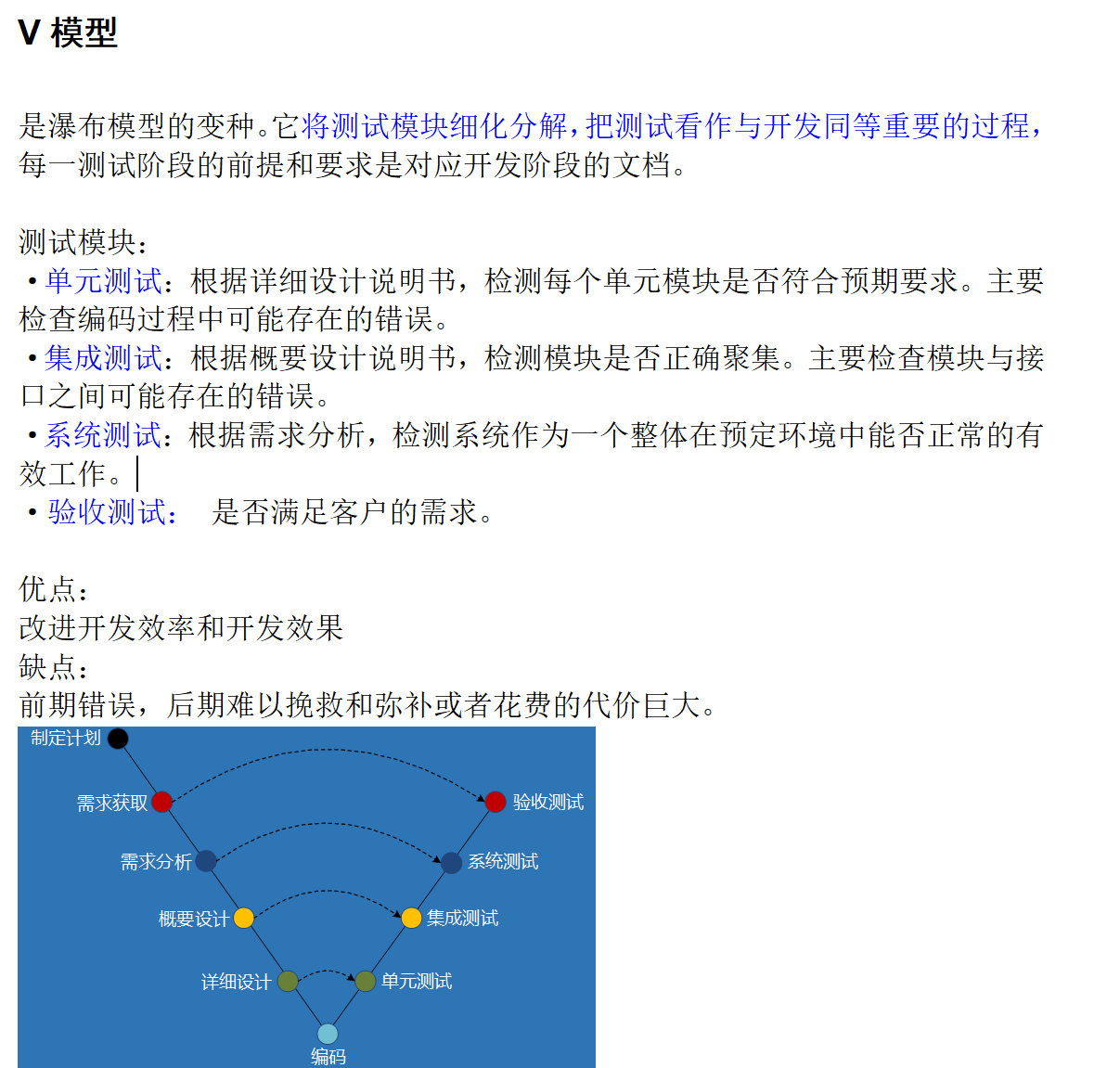
**4.传统模型**  
4.1瀑布模型  


4.2演化模型  

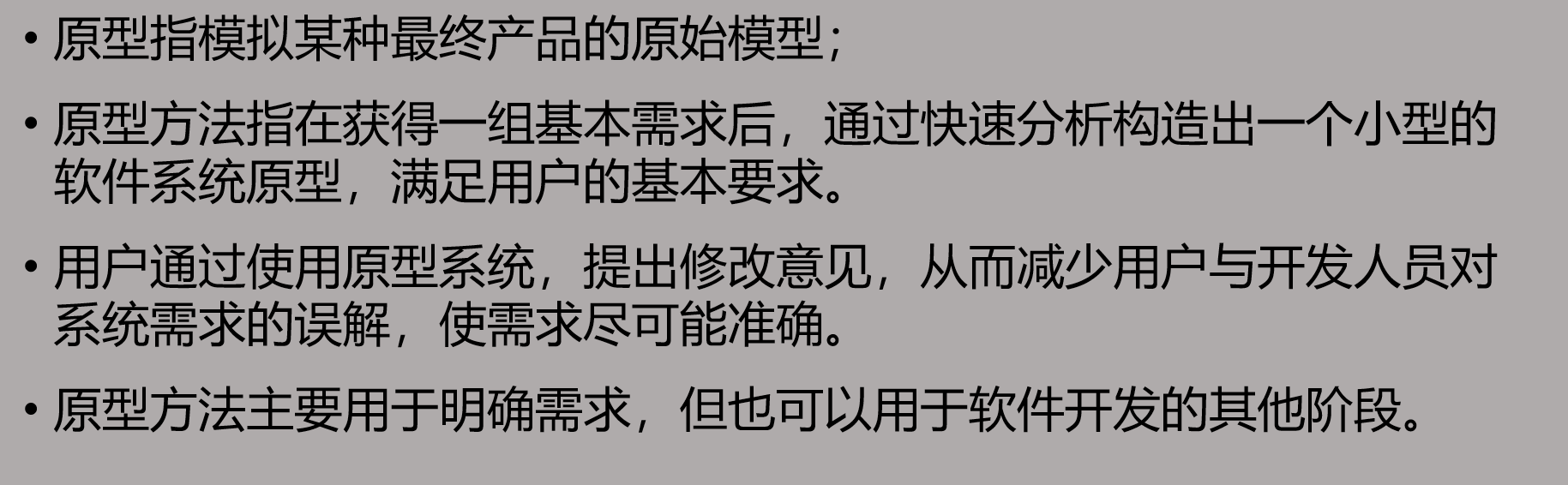

4.3增量模型

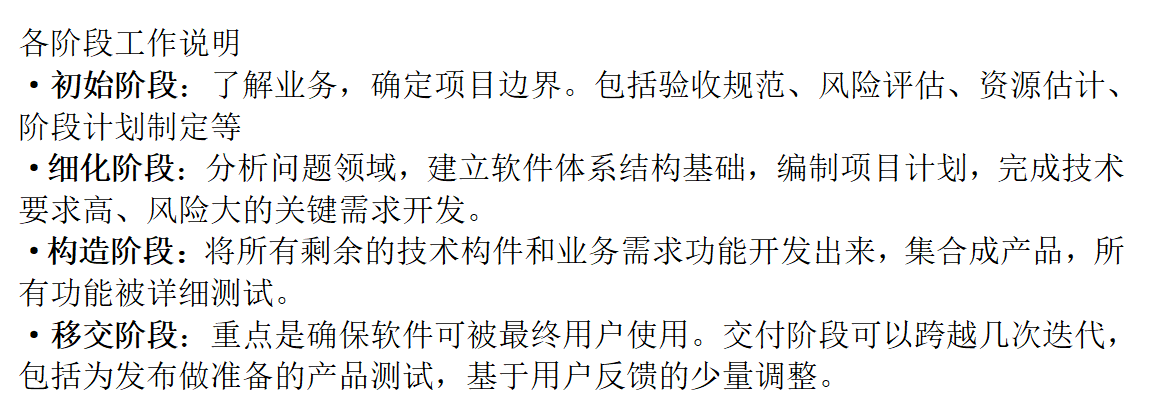
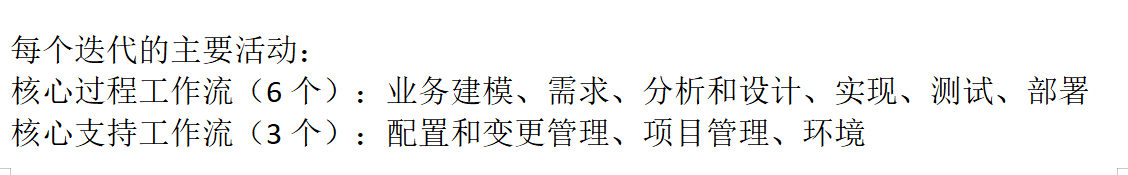
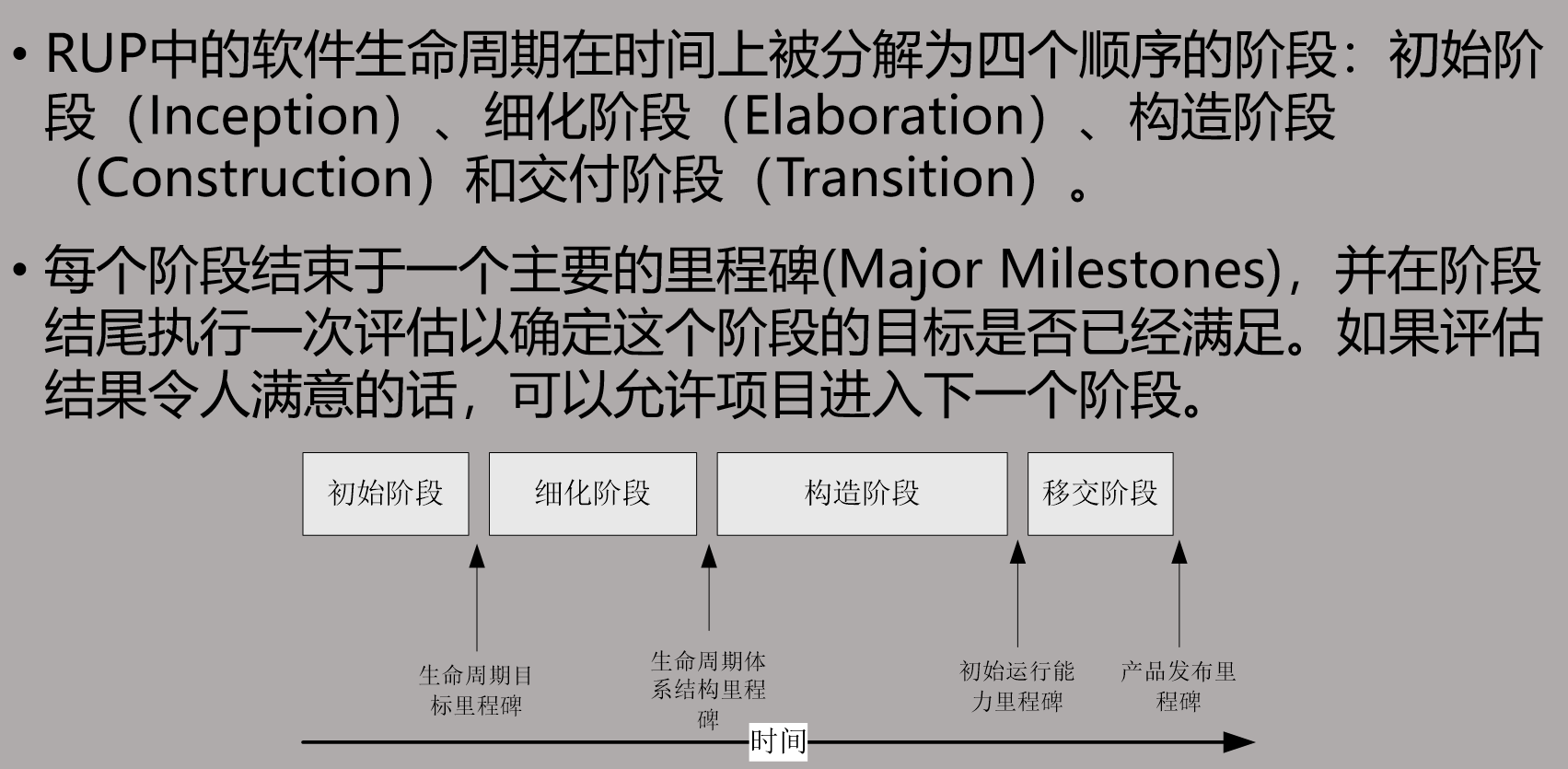
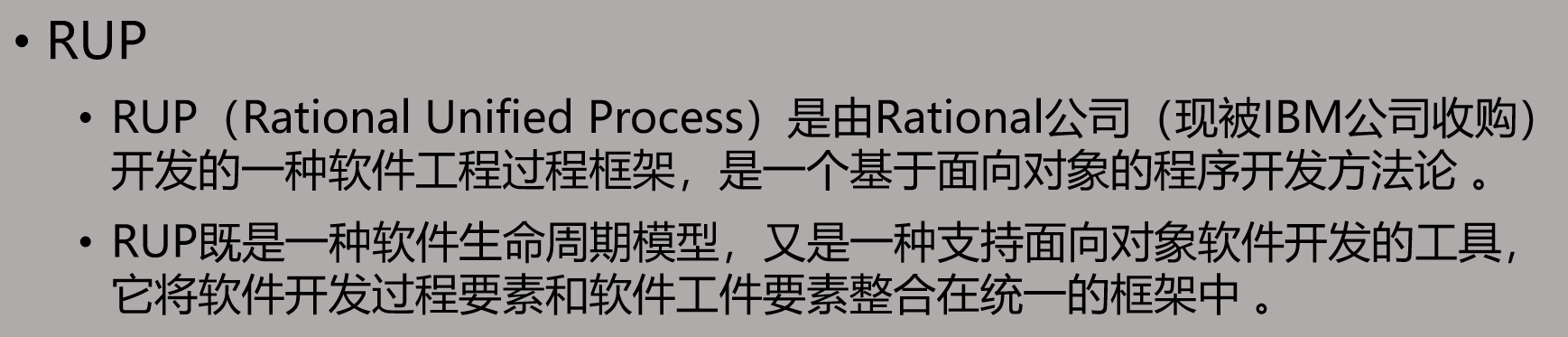


4.4喷泉模型  




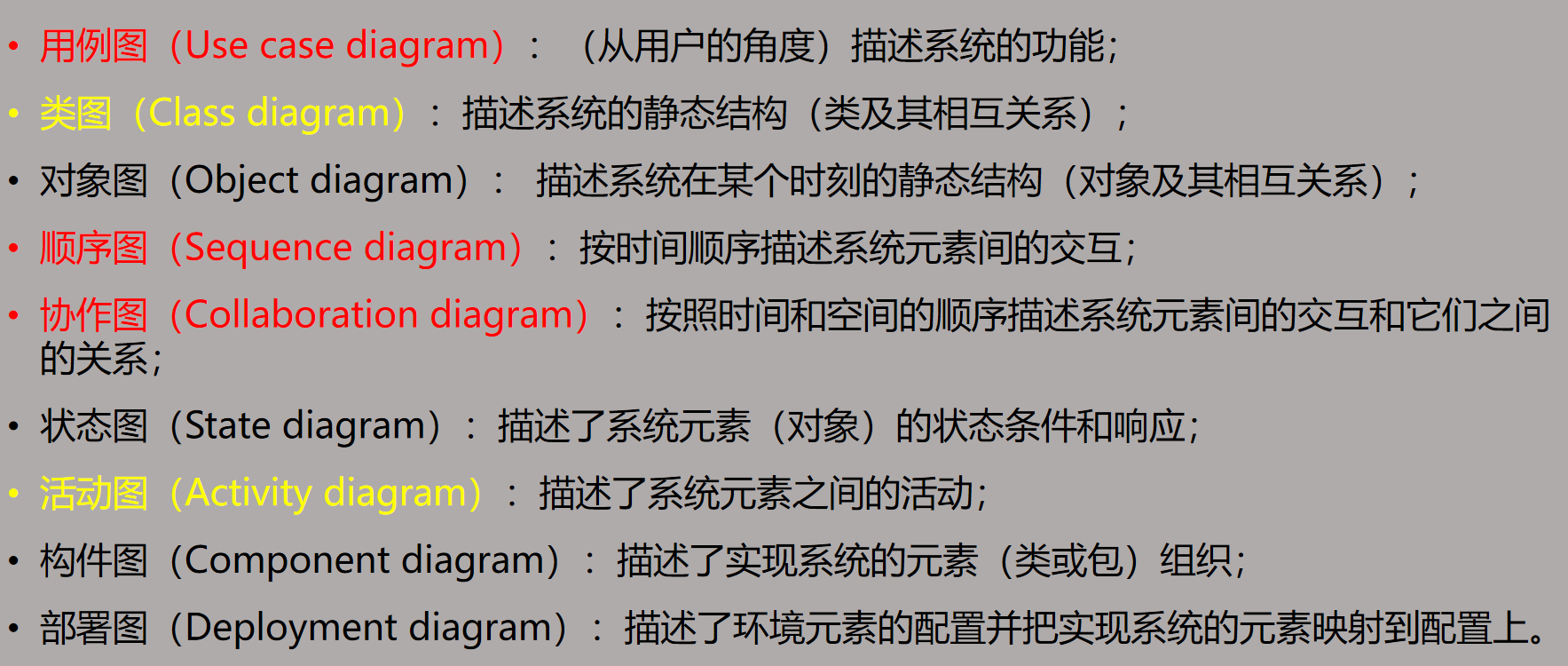
  
原型方法：



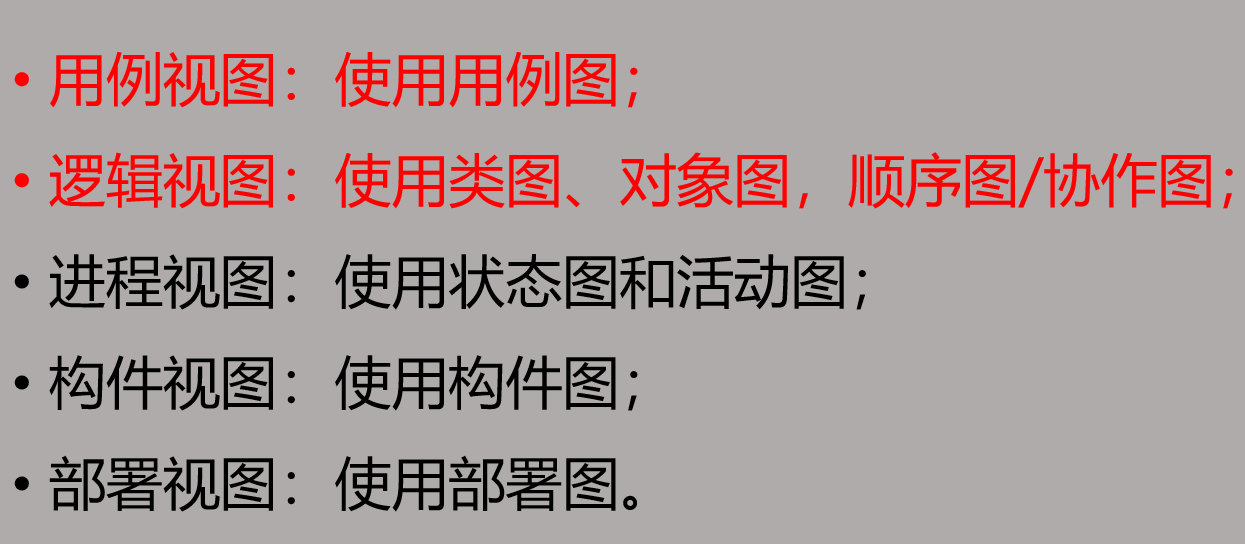
**新型软件生命周期模型：**

* 1. 需求分析的对象、任务和目标、原则
     1. **对象：用户需求**
     2. **任务：准确定义新系统的目标，编制《需求规格说明书》**
     3. **目标：**借助**当前系统的逻辑模型**导出**目标系统的逻辑模型**，解决目标系统的**“做什么”**的问题。
     4. 原则：
        1. **操作性原则：**信息域必须被表示和被理解；功能必须被定义；行为必须被表示
        2. **工程化原则：**
           1. 正确理解问题，之后再**建立分析模型**；
           2. 记录每个需求的起源及原因，保证需求的**可回溯性**；
           3. 开发人机交互过程的原型；
           4. 给需求赋予优先级
           5. 删除歧义性

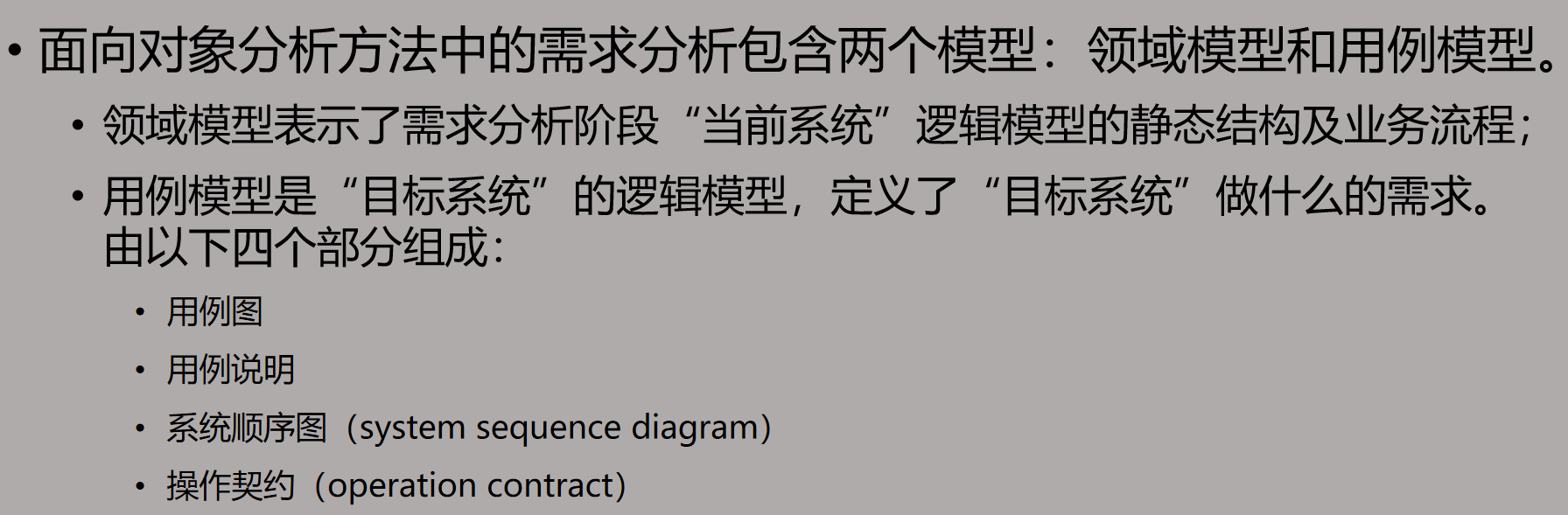
UML的9个基本图：



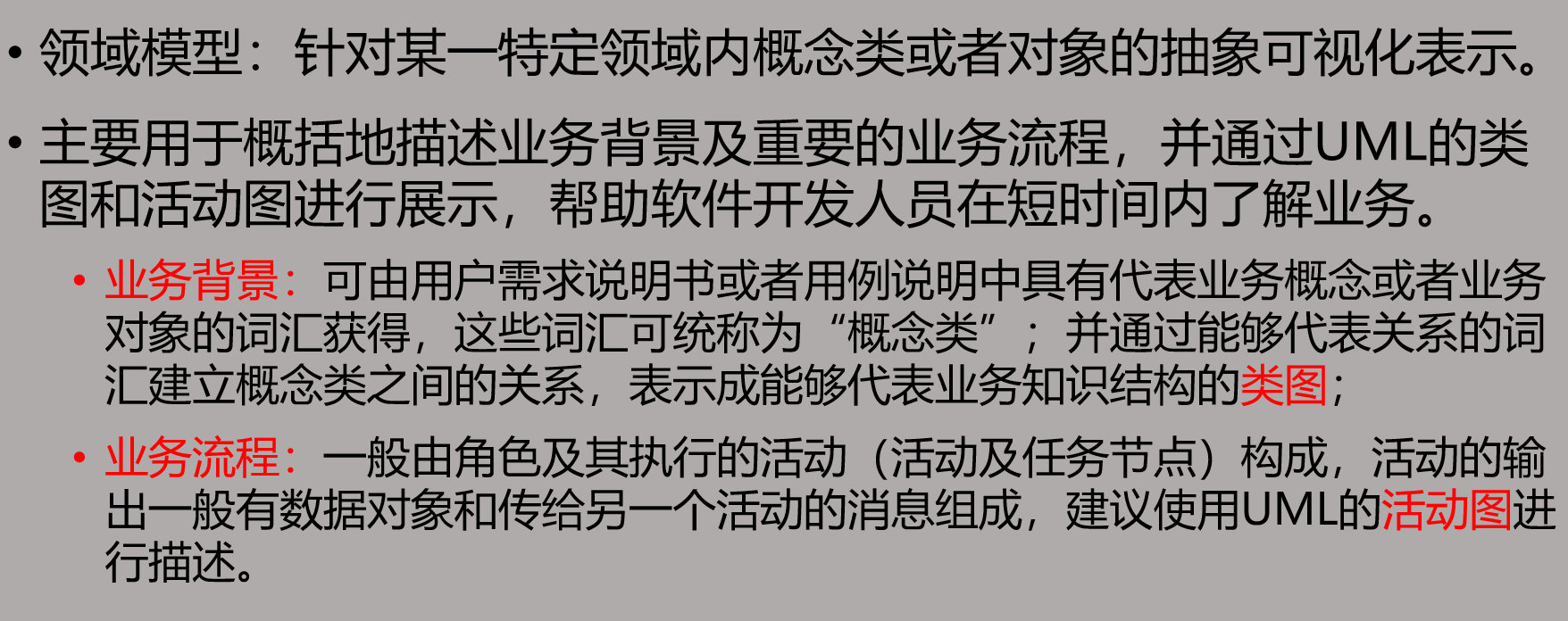
UML视图与图的关系：

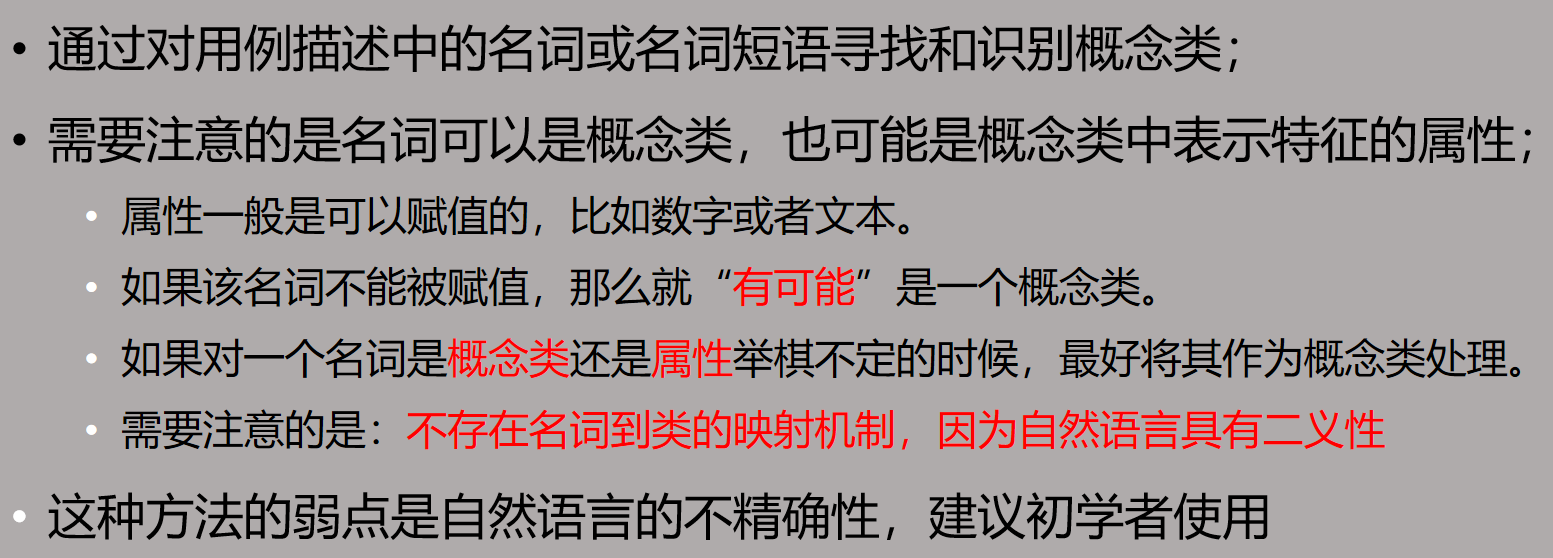


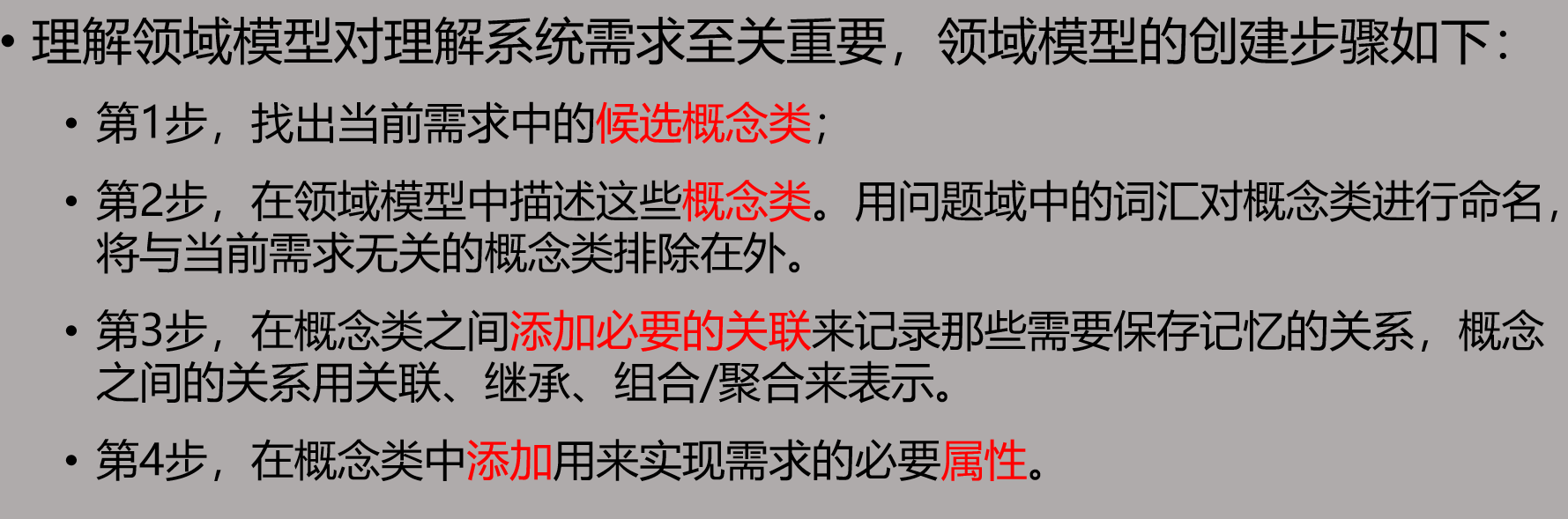
**面向对象的需求分析建模：**



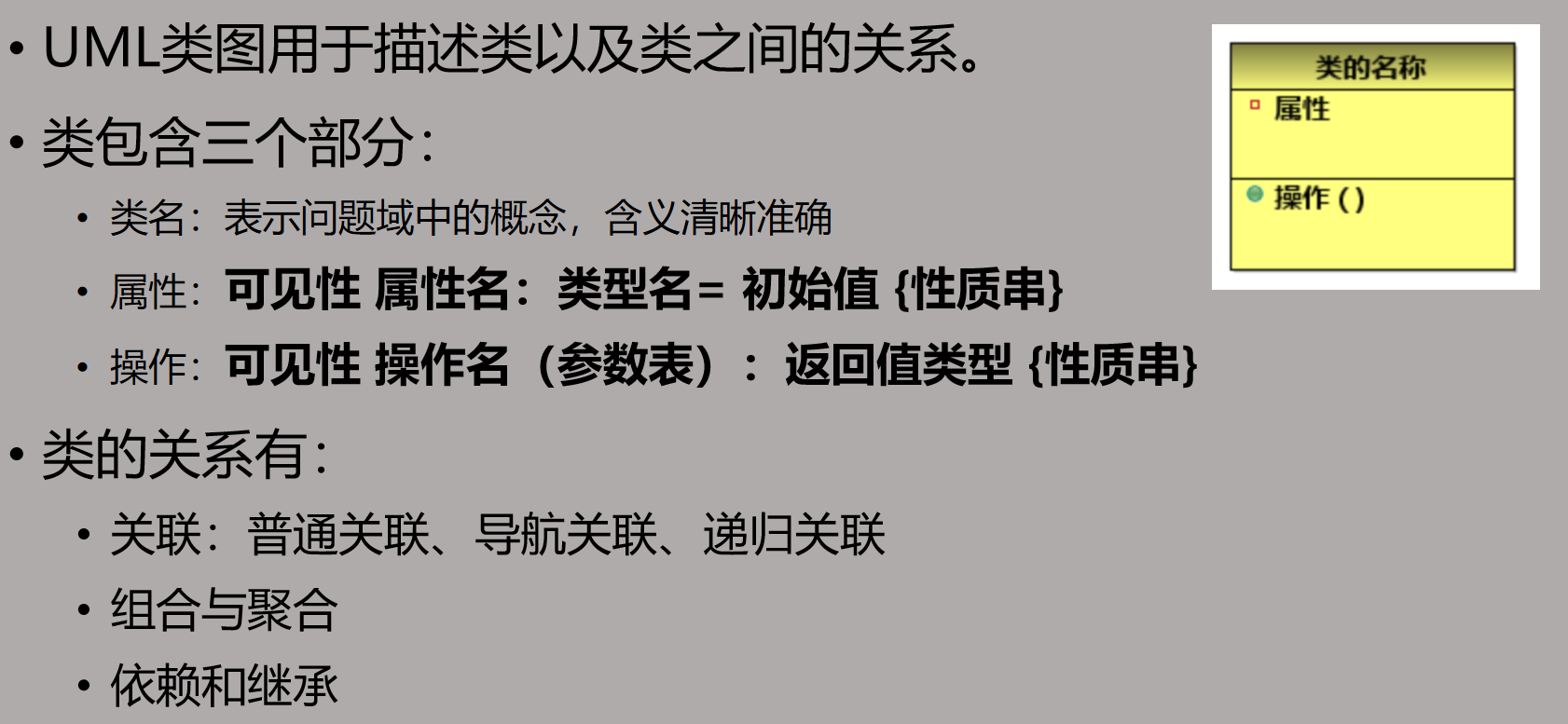
**领域模型：**



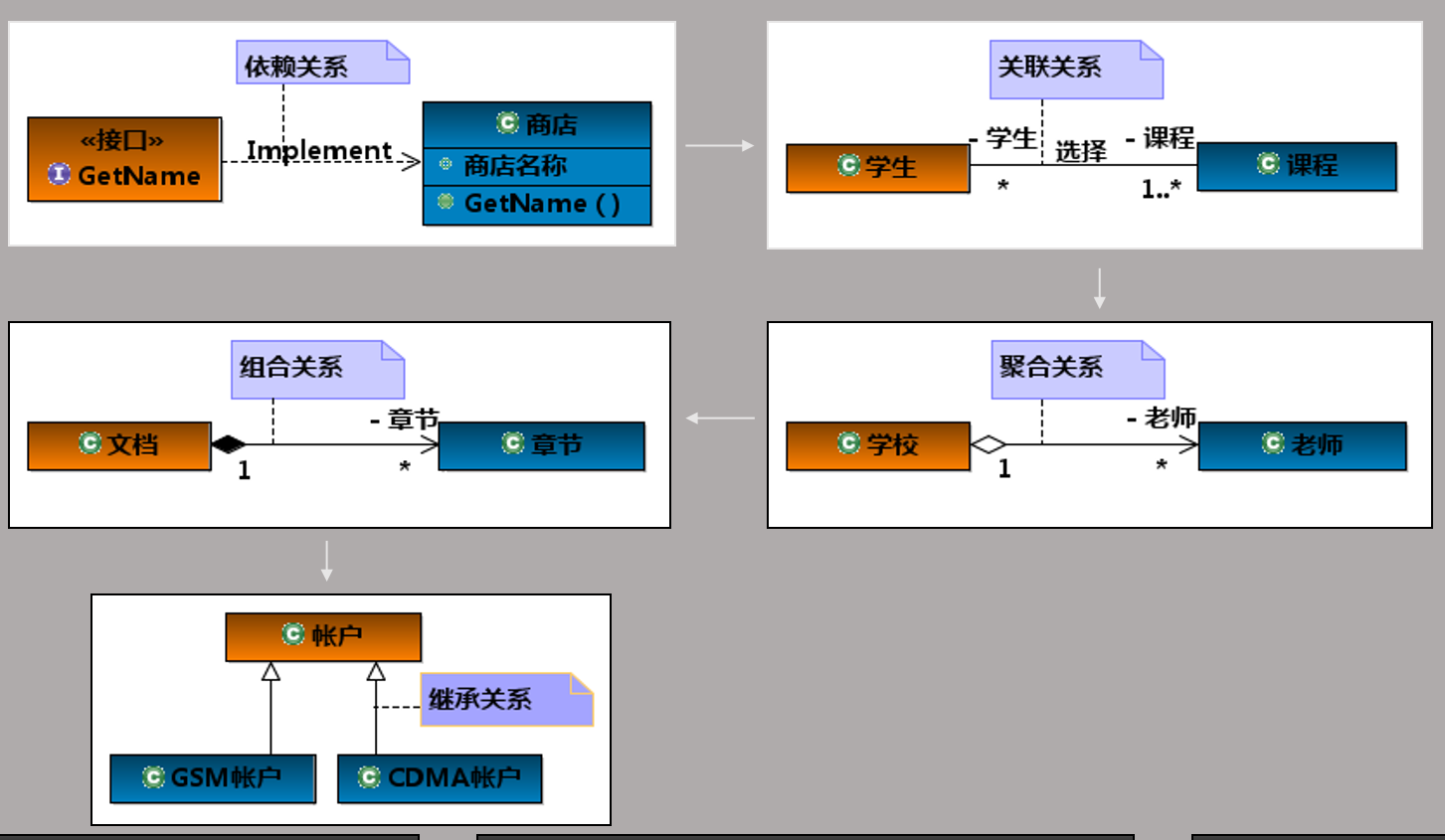




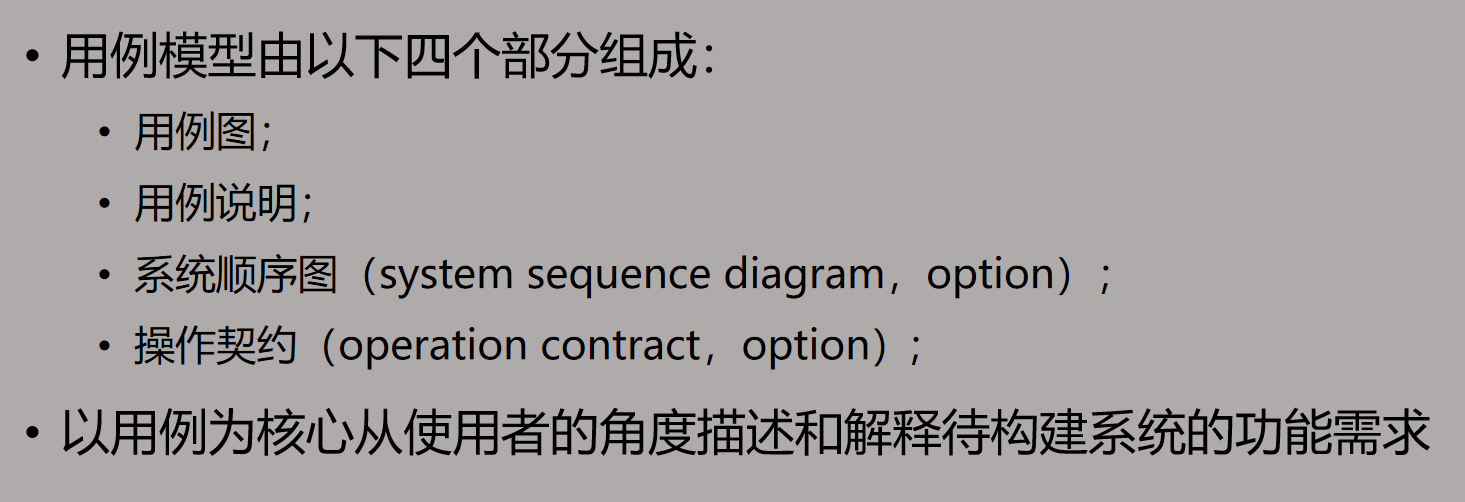
**UML 类图的组成**

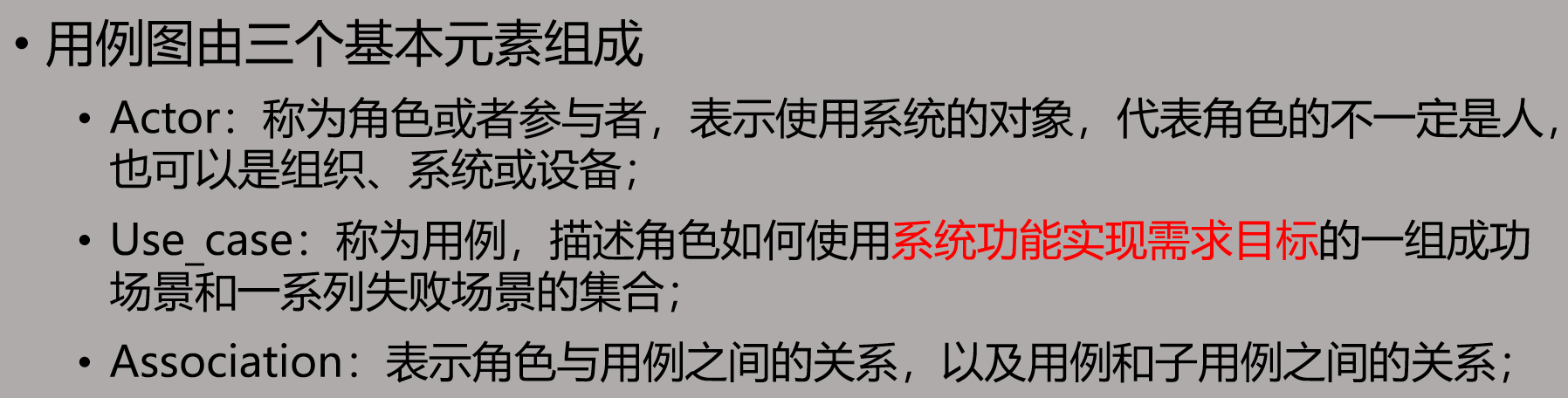


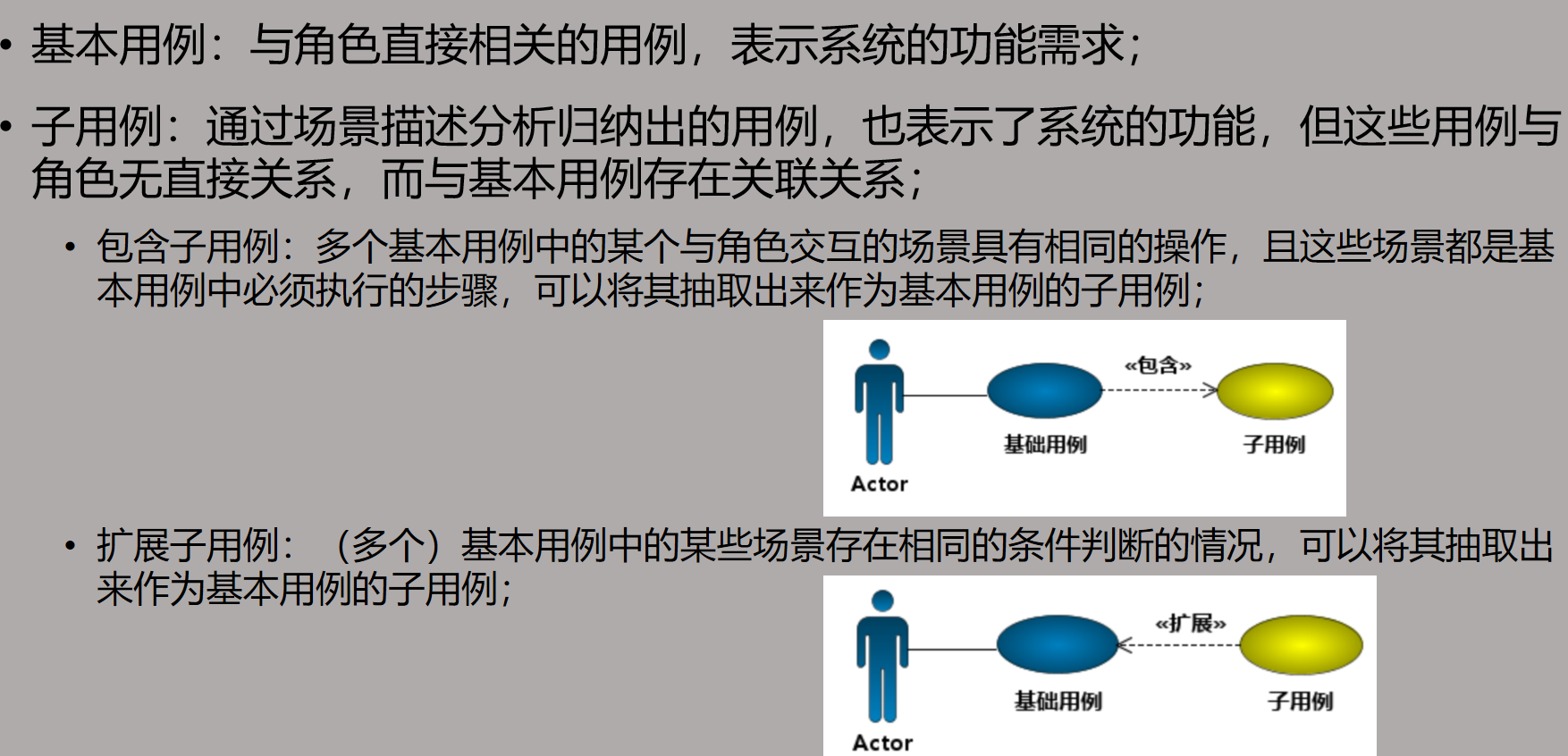
* 1. 类的另一种分类：
     1. 依赖关系：
        1. 类A 把 类B的实例 作为 方法里的参数使用；
        2. 类A的某个方法里 使用了 类B的实例作为局部变量；
        3. 类A 调用了 类B的静态方法
     2. 关联关系
        1. 一个类的属性声明另一个类的对象，或者定义另一个类的引用
     3. **聚合关系**
     4. **组合关系**
        1. **注意：聚合和组合的关系**
           1. **作图时，作为整体的概念里带有菱形，作为部分的概念类带有箭头——分清整体和部分**
           2. **组合：黑菱形；聚合：白菱形**
           3. **组合：整体消失，部分也不存在**
           4. **聚合：整体消失，部分依然存在**
     5. 继承关系



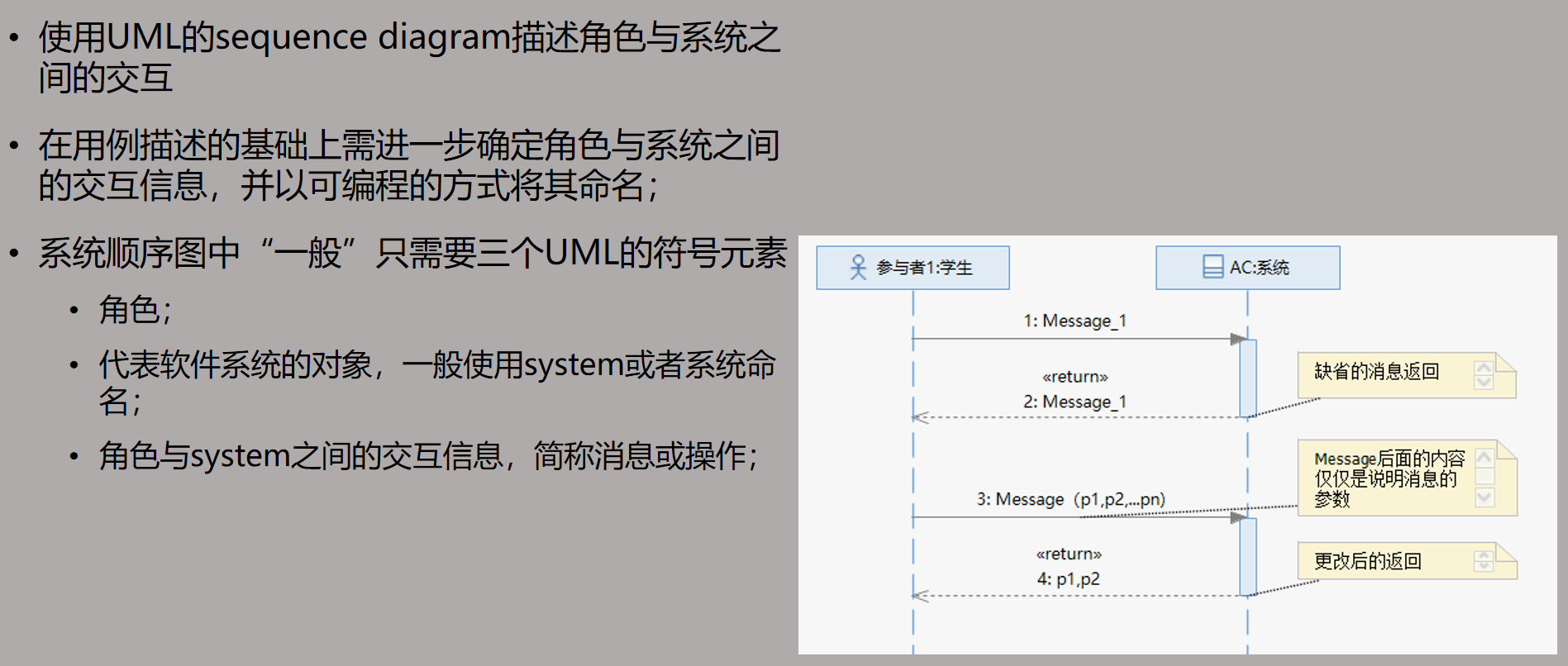
用例模型：



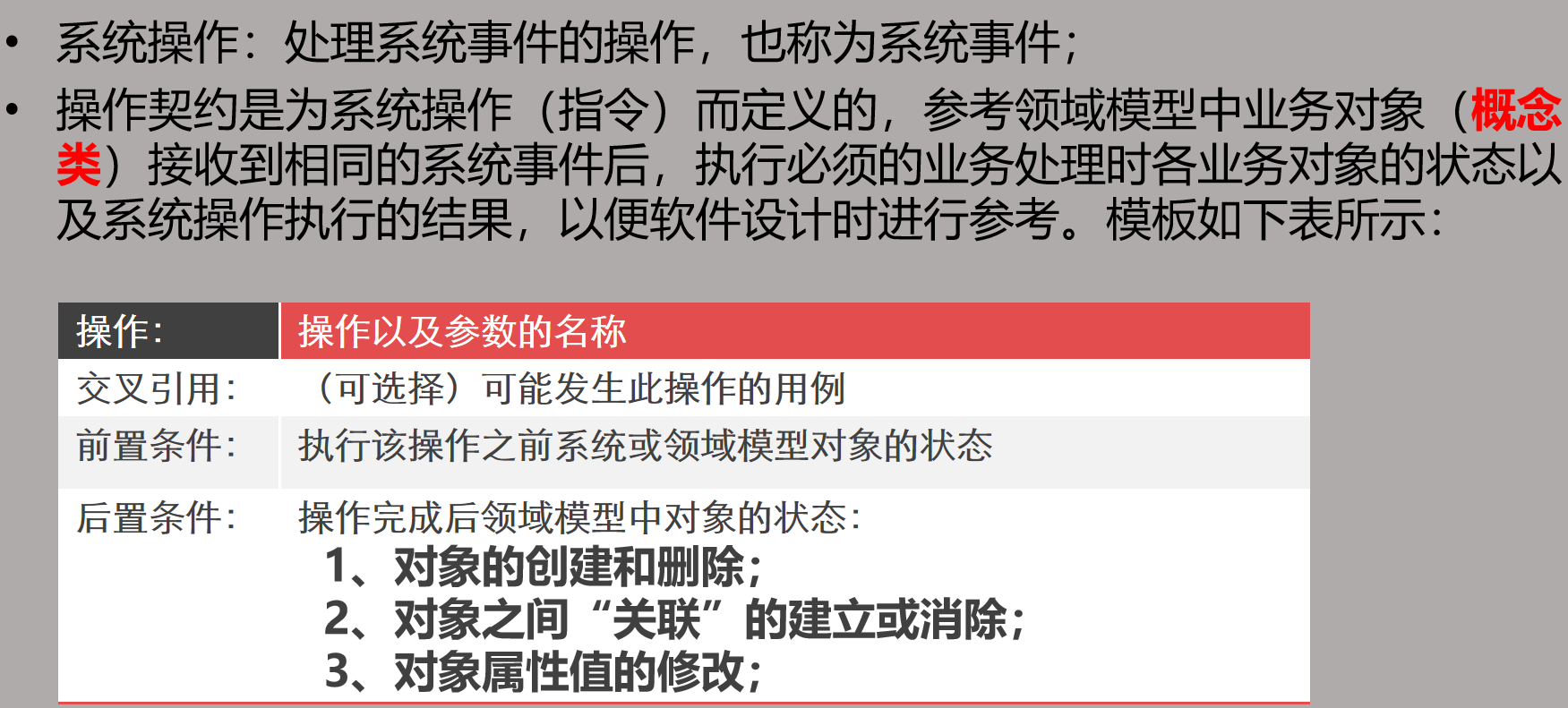


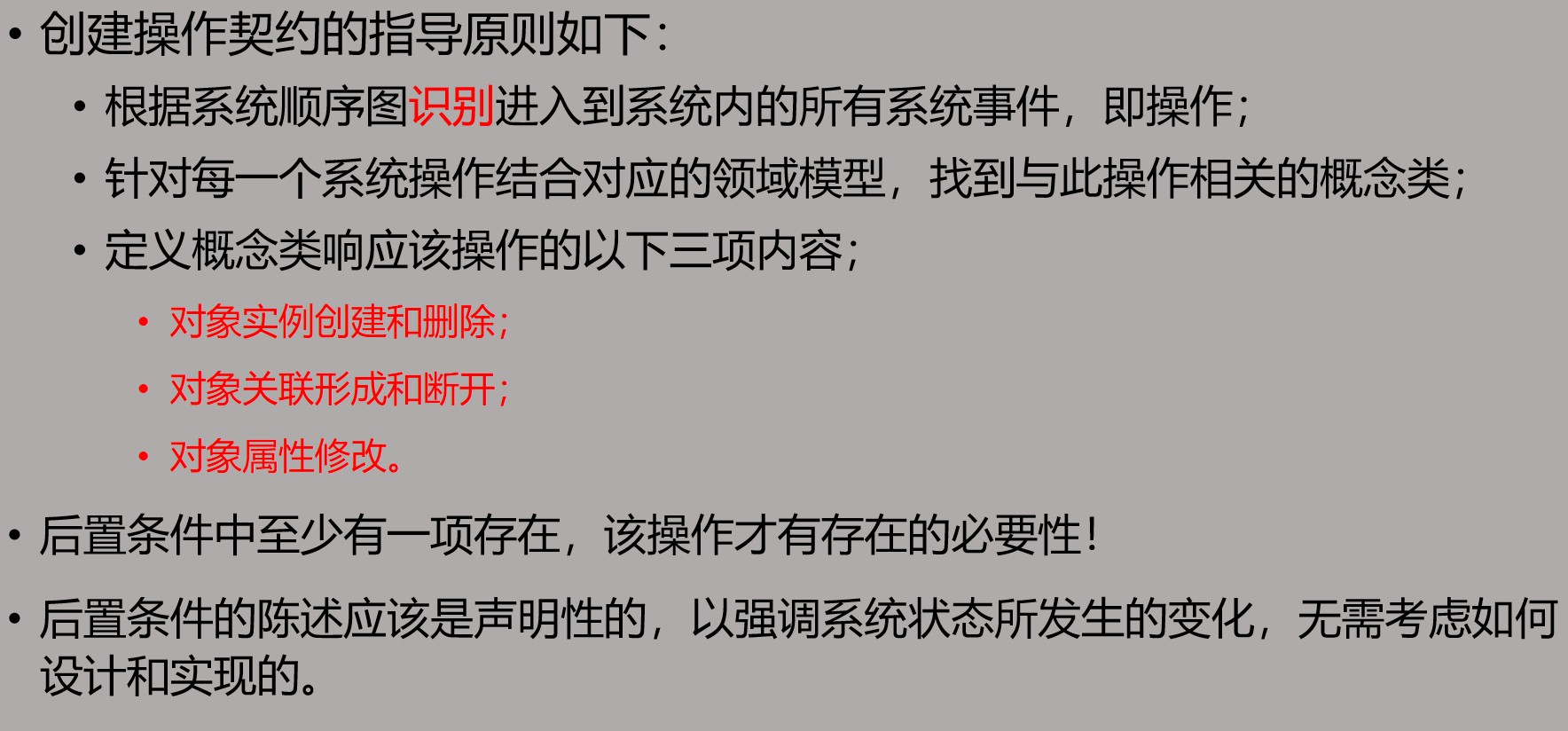


系统顺序图（SSD）：



操作契约：





数据流图：

