第一部分：

1、软件工程定义：

软件工程是指导计算机软件开发和维护的工程学科。它采用工程化的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明是正确的管理方法和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护它。

2、软件工程的三要素：

（1）方法：**为软件开发提供“如何做”的技术**

（2）工具：**为软件开发提供了自动的或半自动的软件支撑环境**

（3）过程：**将软件工程的方法和工具综合起来以达到合理、及时的效果**

**3、软件工程的目标：**

**运用先进的软件开发技术和管理方法来提高软件的质量和生产率，也就是要以短的周期，低的成本生产出高质量软件产品，并最终实现软件的工业化生产。**

4、软件危机的概念：

**就是指在软件开发和软件维护过程中所存在的一系列严重问题**

**5、软件危机的背景：**

**随着计算机应用在各行各业，解决问题的规模和难度在逐渐增加，由于软件本身的特点和软件开发方法等问题，软件的发展速度远远滞后于硬件发展速度，不能满足社会日益增长的软件需求**

**6、软件危机的表现：**

**（1）软件开发没有真正的计划性，对软件开发进度和软件开发成本的估计常常很不准确，计划的制定带有很大的盲目因素。**

**（2）对于软件需求信息的获取常常不充分，软件产品往往不能真正地满足用户的实际需求。**

**（3）缺乏良好的软件质量评测手段，从而导致软件产品的质量常常得不到保证。**

**（4）对于软件的可理解性、可维护性认识不够；软件的可复用性、可维护性不如人意。**

**（5）软件开发过程没有实现“规范化”，缺乏必要的文档资料或者文档资料不合格、不准确，难以进行专业维护。**

**（6）软件开发的人力成本持续上升。**

**（7）缺乏自动化的软件开发技术，软件开发的生产率依然低下，远远满足不了急剧增长的软件需求**

1. **软件危机的解决方法：**

**（1）从管理层面上考虑，应当注意推广和使用在实践中总结出来的开发软件的成功的技术和方法，并且探索更好的、更有效的技术和方法，注意积累软件开发过程中的经验数据财富，逐步消除在计算机系统早期发展阶段形成的一些错误概念和做法。**

（2）**从技术角度考虑，应当开发和使用更好的软件开发工具，提高软件开发效率和开发工作过程的规范化程度**

第二部分

1、需求分析的定义：

需求分析是指对项目或系统中所需要的功能、性能和其他特定要求进行详细而系统的研究和识别的过程。

2、需求获取方法：

（1）召开相关会议，走访相关人员

（2）设计调查问卷，填写和回收问卷

（3）现场观摩，了解业务的过程和步骤，掌握业务工作细节

（4）阅读和分析业务相关文档和资料

3、需求描述方法：

（1）功能建模：数据流图  
（2）数据建模：ER图

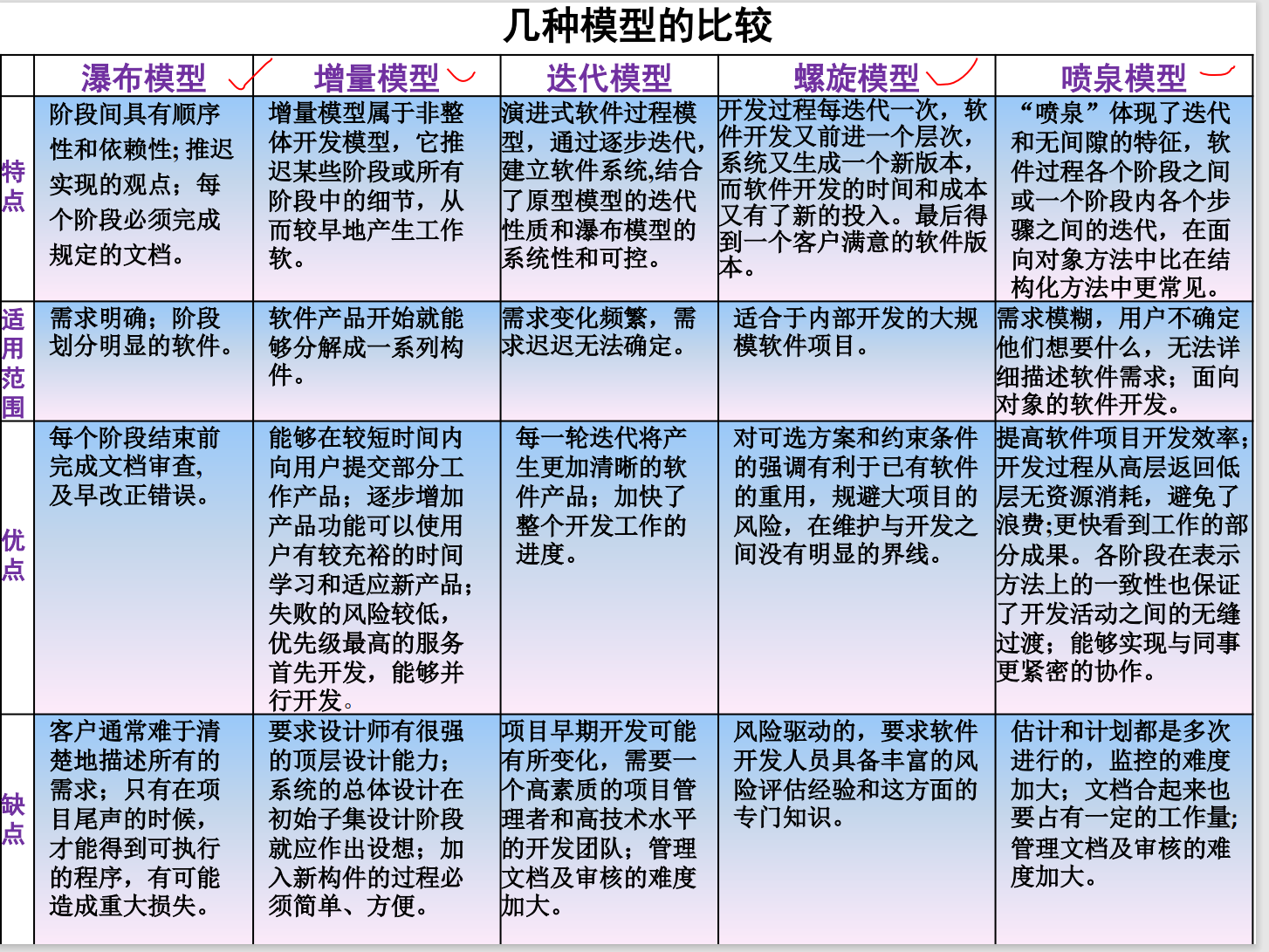
（3）行为建模：状态转换图（不考）

（4）数据字典：定义式和 Warnier 图

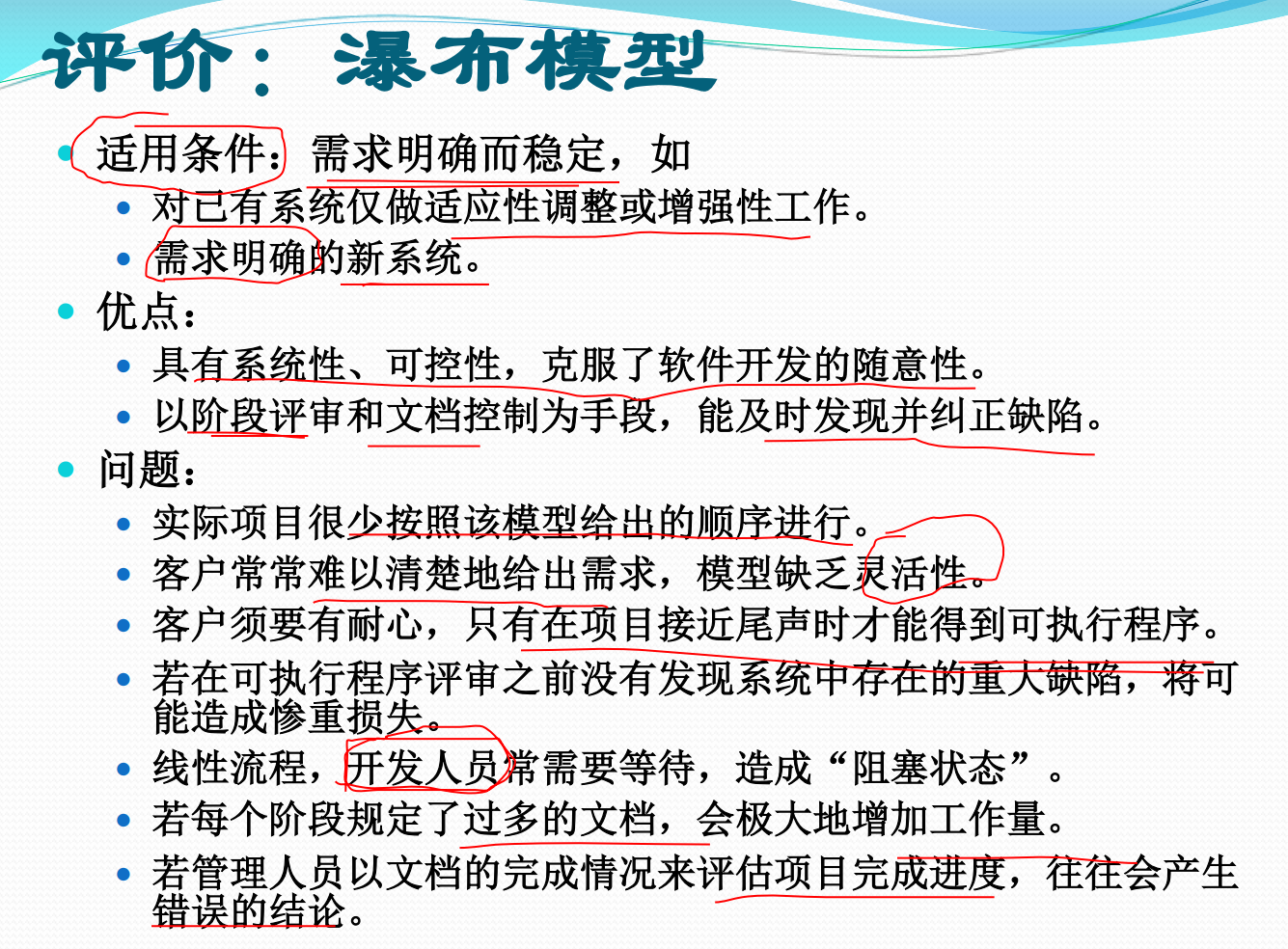
第三部分：

1、软件生存期模型概念：从项目需求定义直至软件运行维护为止，跨越整个生命周期的系统开发、运行和维护所实施的全部过程、活动和任务的结构框架。

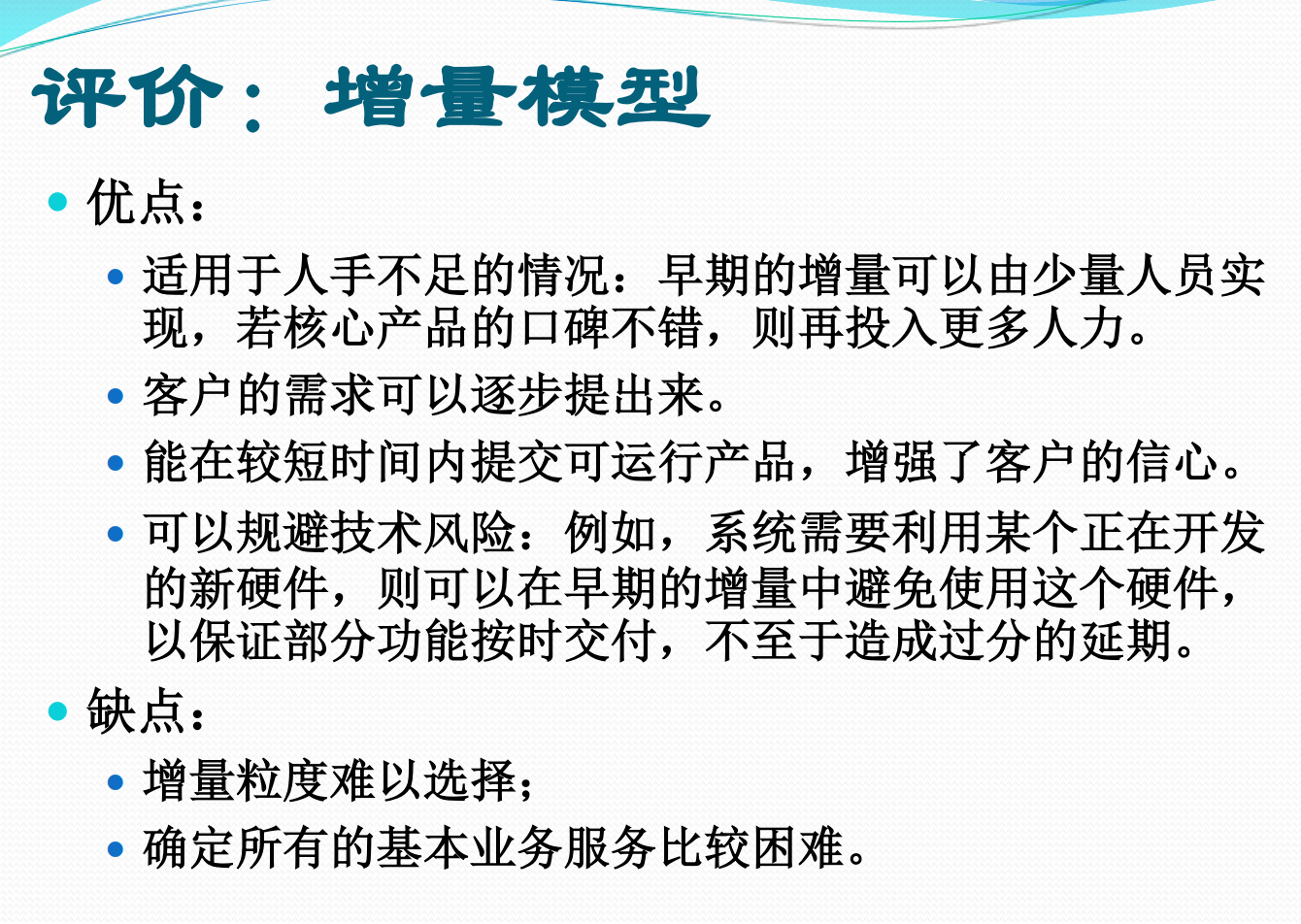
2、6中生存期模型：

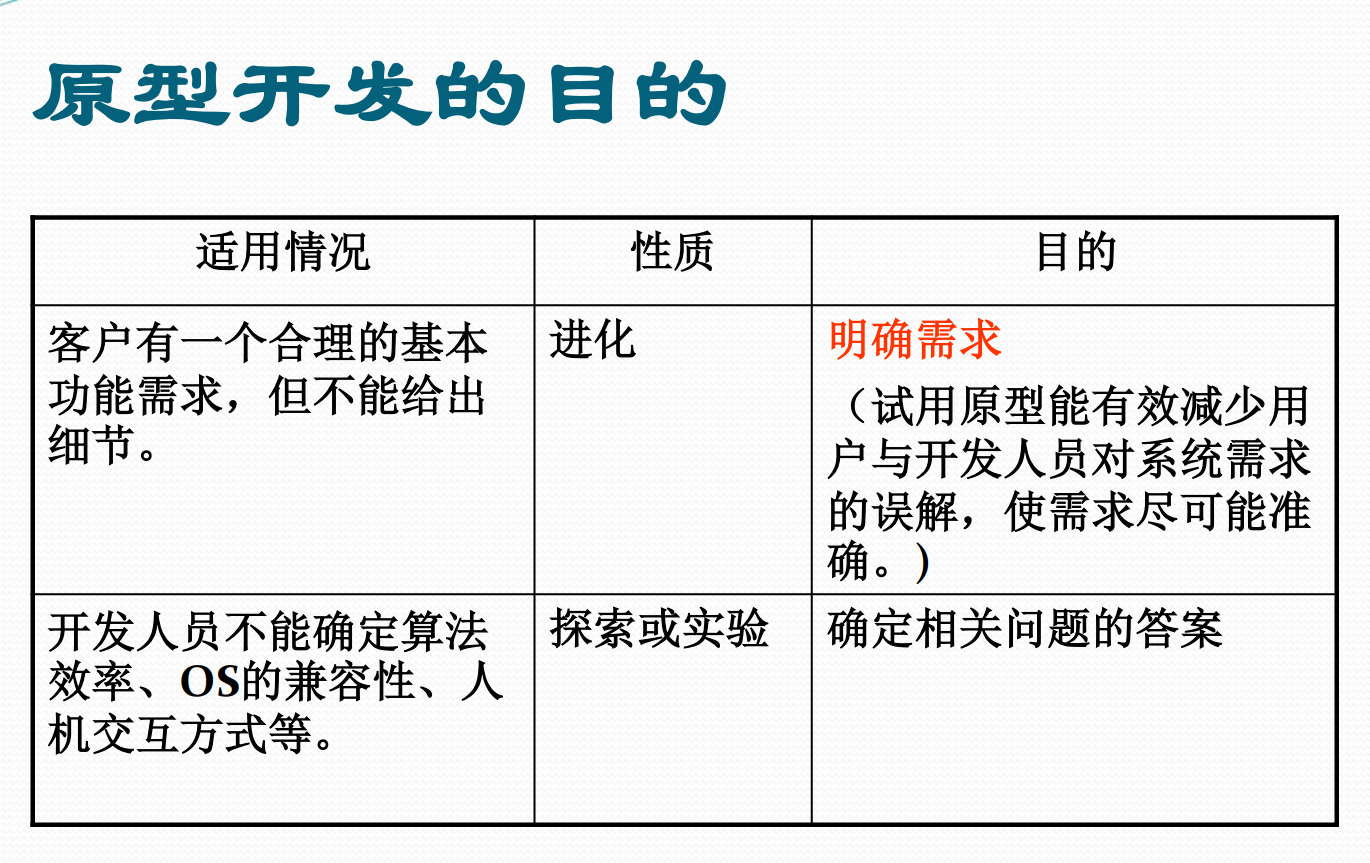


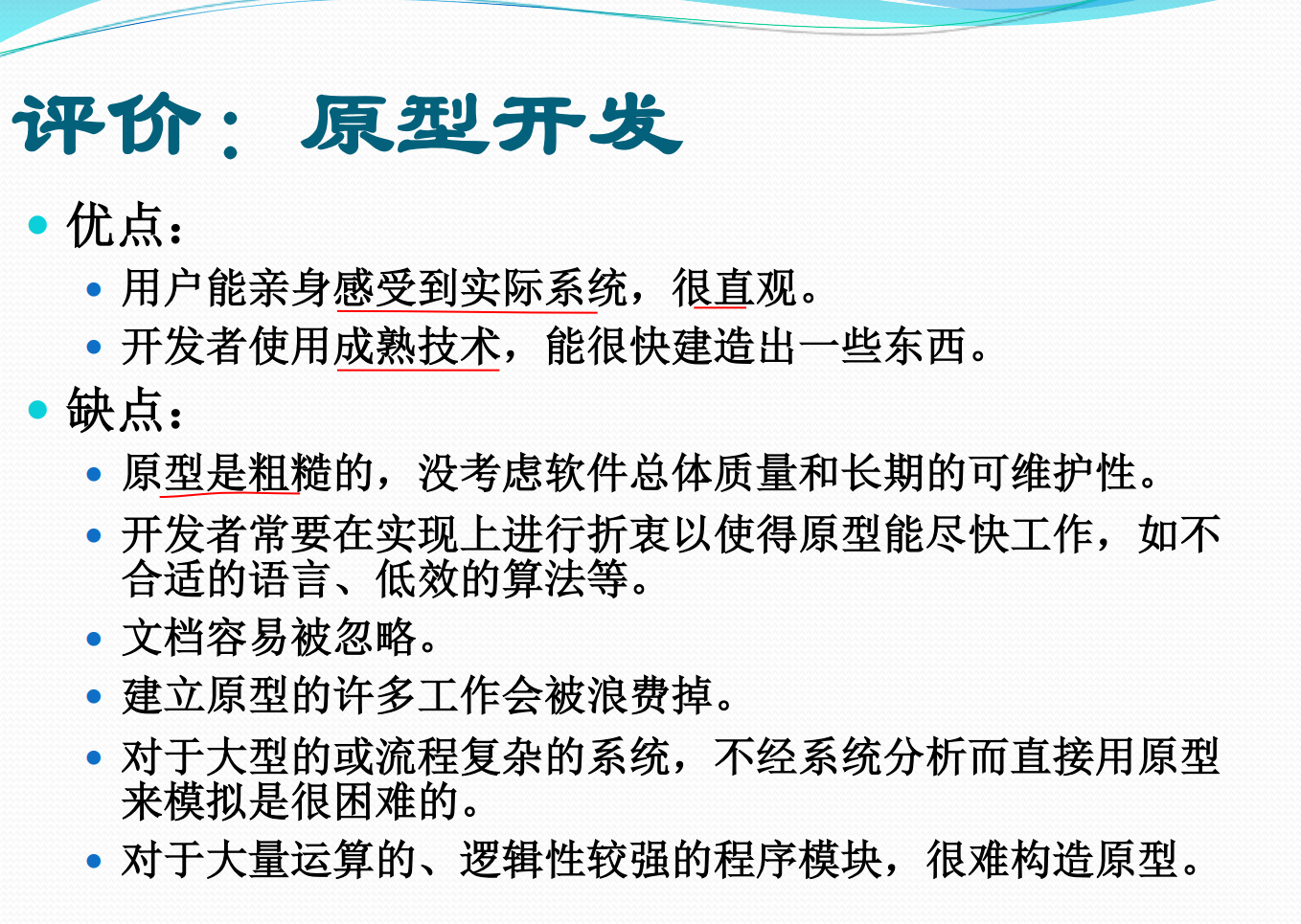
（1）瀑布模型：（2）v模型是瀑布模型的变体



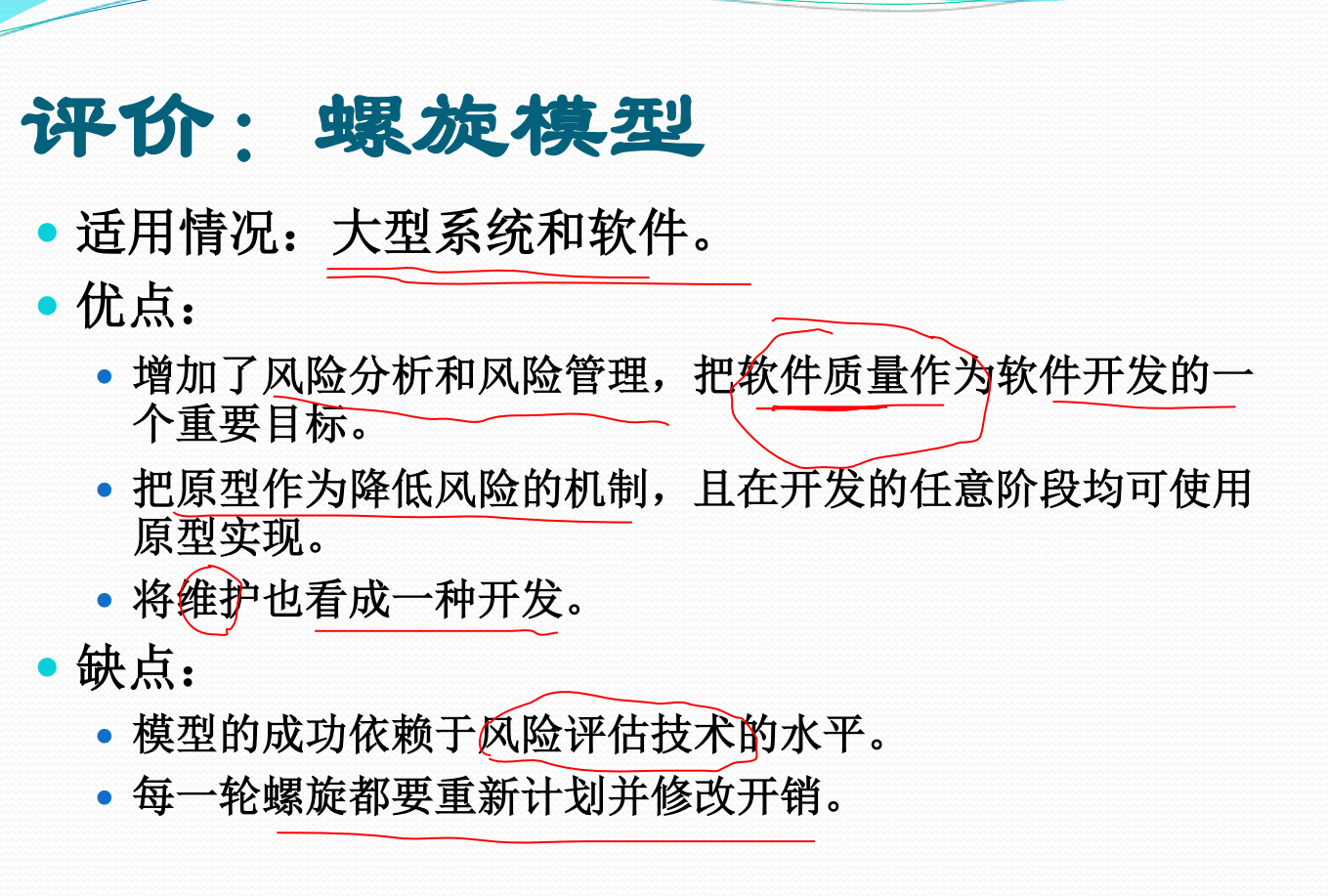
（3）增量模型：



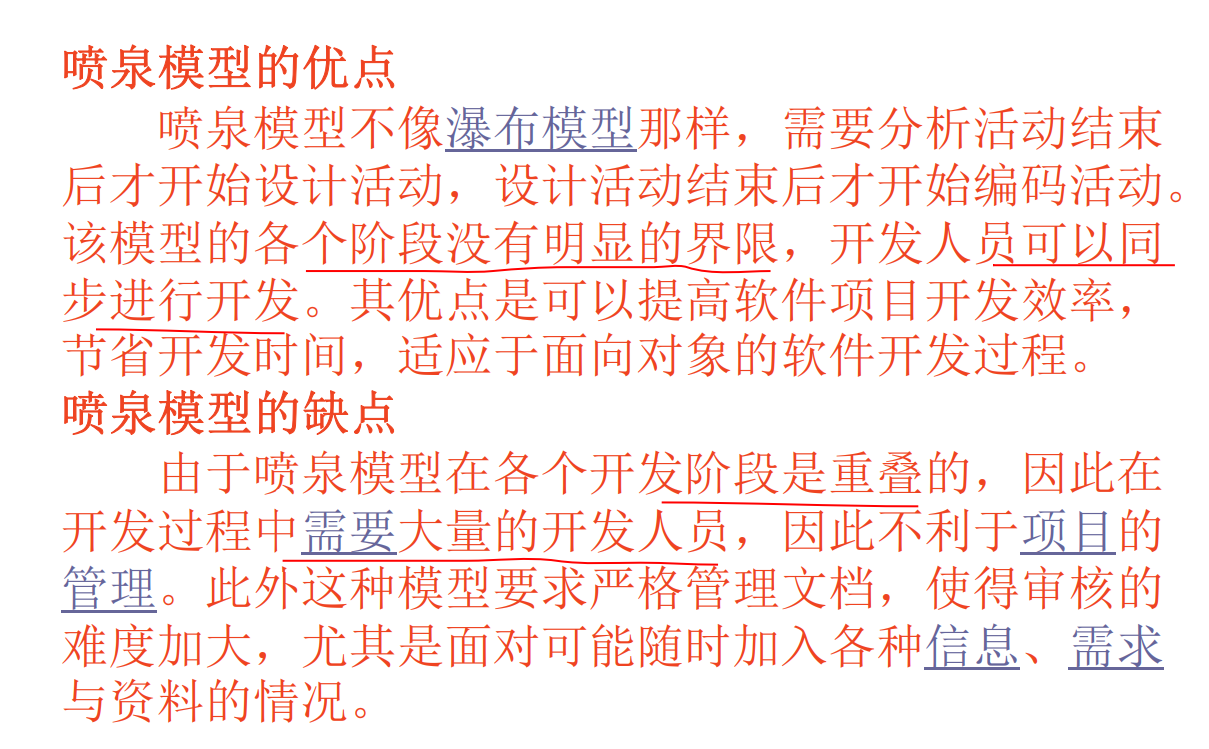
（4）原型开发模型：



（5）螺旋模型：



（6）喷泉模型：



第四部分：

1、功能建模：数据流图

2、数据建模：ER 图

3、数据字典：定义式和Warnier 图

4、加工规格：决策表和决策树

第五部分：

1、软件设计的原则：

（1）**分而治之**：**将大型复杂的问题分解为许多容易解决的小问题。**

（2）模块独立性：是指软件系统中每个模块只涉及软件要求的具体的子功能。

补充（好像会考概念，老师第一次划重点的时候提过）：

耦合:是模块之间的互相连接的紧密程度的度量。

内聚:是模块功能强度(一个模块内部各个元素彼此结合的紧密程度)的度量。

模块独立性比较强的模块应是高内聚低耦合的模块。

2、软件结构图

第六部分：

1、关联关系：整体和部分之间的结构关系

2、泛化关系：类和特殊类之间的继承关系

注意银行的例子

3、用例图

4、类图

5、顺序图

第七部分：

1、语句覆盖：使得每个语句都被执行至少一次

2、判定覆盖：**使得程序中的每一个判定取真分支和取假分支至少都通过一次**

**3、条件覆盖：执行足够的测试用例，使得判定中的每个条件的可能取值至少满足一次。**

**4、判定/条件覆盖：使得分支中每个条件取到各种可能的值，并使每个分支取到各种可能的结果**

**5、条件组合覆盖：使得每个判定中条件的各种可能组合都至少出现一次**

**6、路径测试：覆盖被测试对象中的所有可能路径**

**7、控制流图**

**8、环路复杂度**

**（1）等于区域数**

**（2）等于边的个数减去点的个数加2**

**（3）等于判定节点的个数加1**

**第八部分：**

**1、等价类划分法**

2、边界值分析法

第九部分：

1、软件维护方法：  
（1）**改正性维护**

**改正性维护（corrective maintenance）为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应进行的诊断和改正错误的过程。例如，改正性维护可以是改正原来程序中开关使用的错误；解决开发时未能测试各种可能情况带来的问题等。**

**（2）适应性维护**

**随着信息技术的飞速发展，软件运行的外部环境（新的硬、软件配置）或数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化，为了使软件适应这种变化，而修改软件的过程叫做适应性维护（adaptivemaintenance）。例如，需要对已运行的软件进行改造，以适应网络环境或已升级改版的操作系统要求。**

**（3）完善性维护**

**为了满足新的功能与性能要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动叫做完善性维护（perfective maintenance）。例如，完善性维护可能是修改一个计算工资的程序，使其增加新的扣除项目；缩短系统的应答时间，使其达到特定的要求等**

**（4）预防性维护**

**预防性维护（preventive maintenance）是指把今天的方法学用于昨天的系统以满足明天的需要。也就是说，采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编码和测试。**

**第十部分：**

**1、工作量等于规模/生产率**

**2、成本等于工作量×人工价**

**3、关键路径计算**