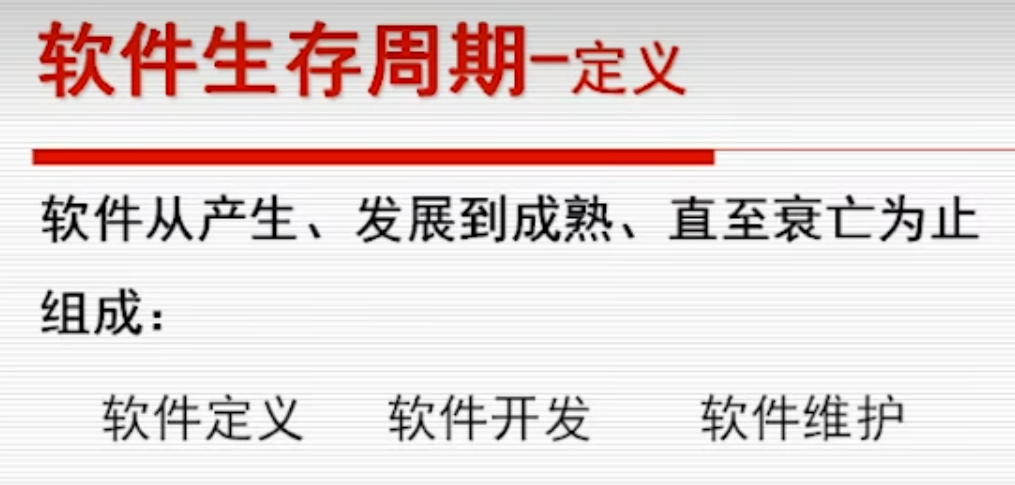
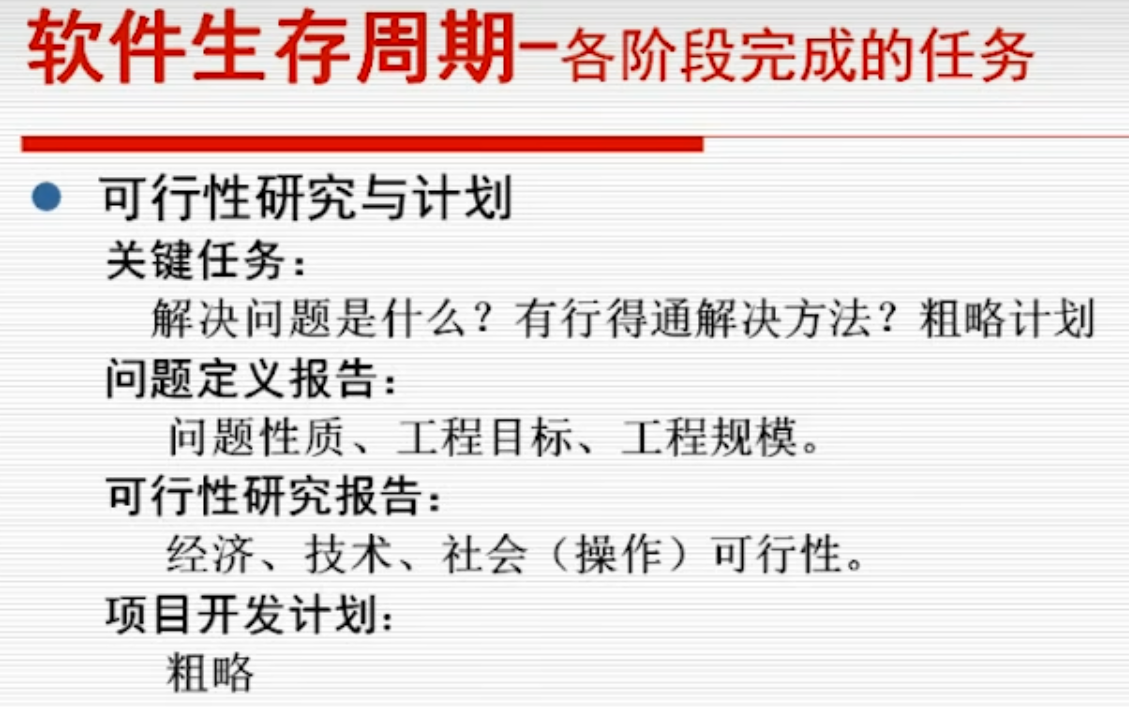
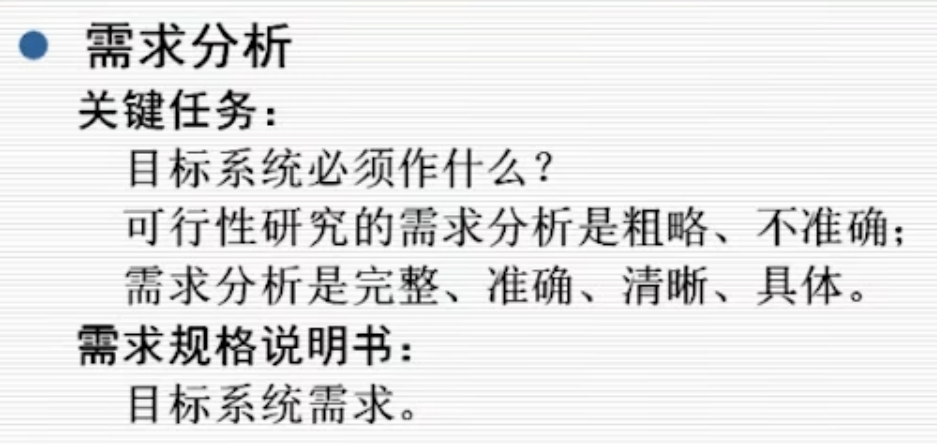
1、软件工程定义：把系统化、规范化、可度量的途径应用于软件开发运行和维护过程中；研究其实现途径。

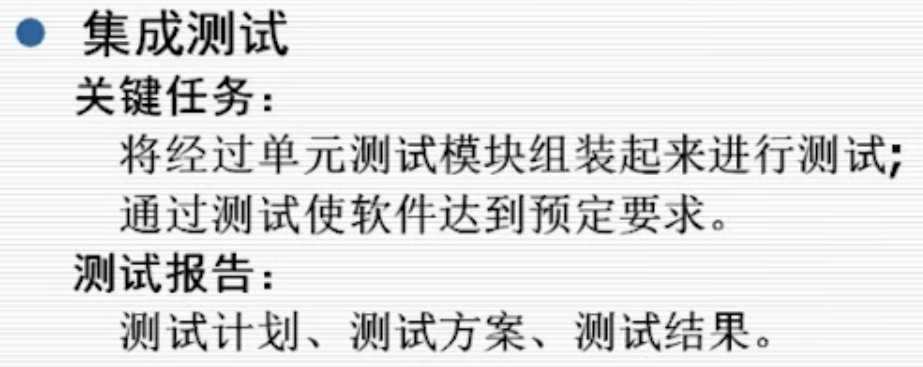
2、

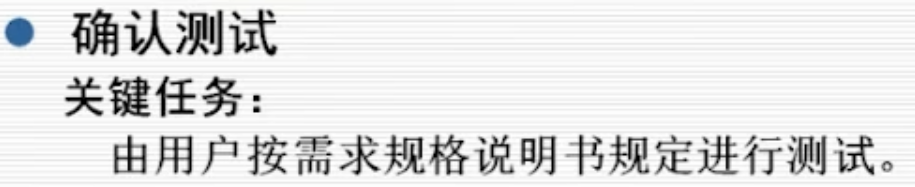
3、

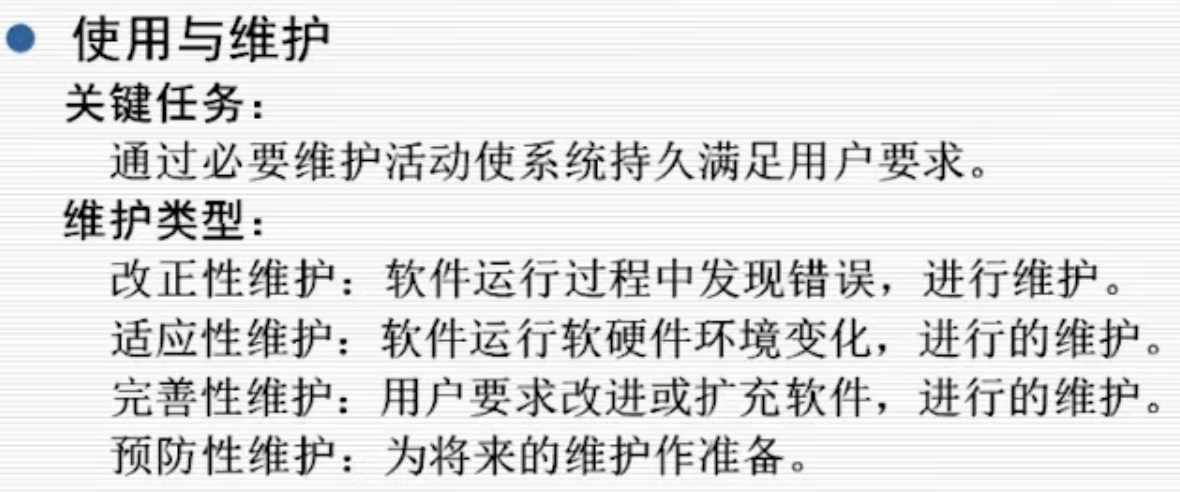
4、

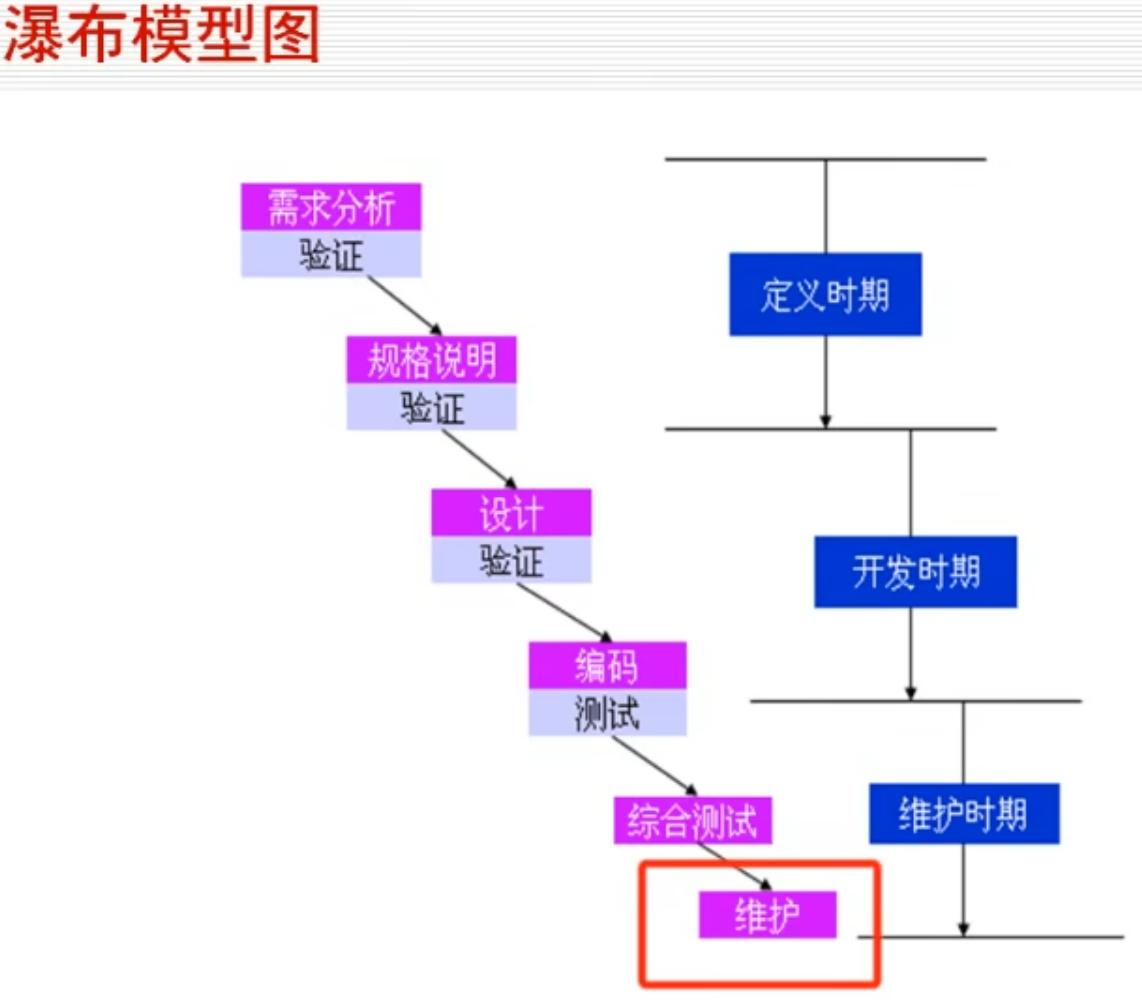
5、

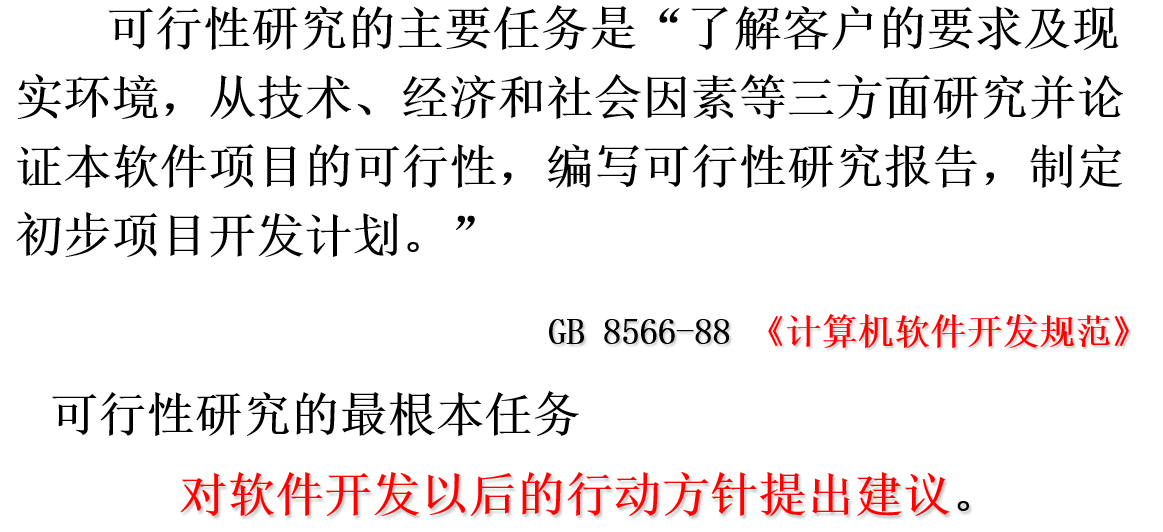
6、

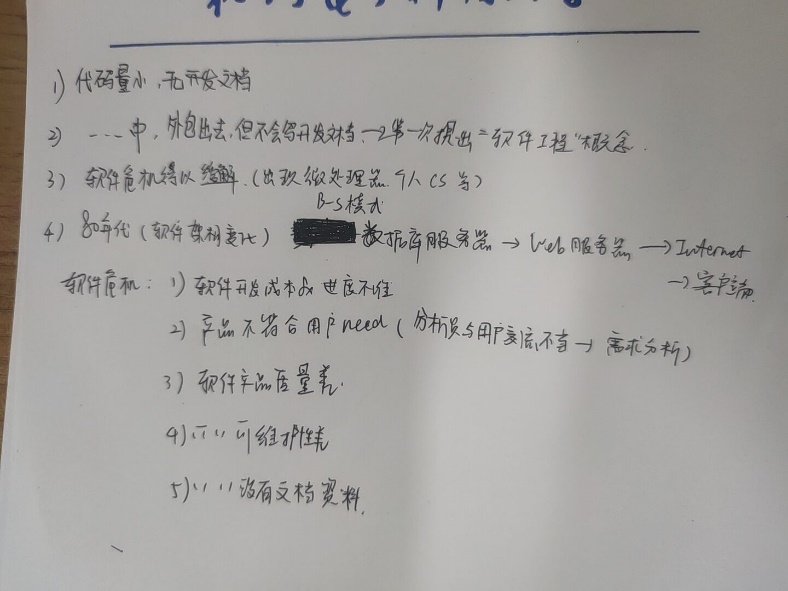
7、

8、

9、

10、

11、

12、

13、投资回收期就是使累计的经济效益等于最初的投资费用所需的时间。

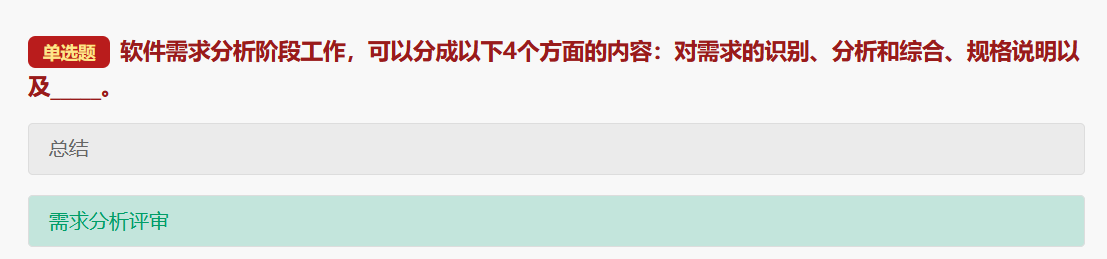
14、笔记：<https://juejin.cn/user/937210532888781/search?search_type=0>

15、

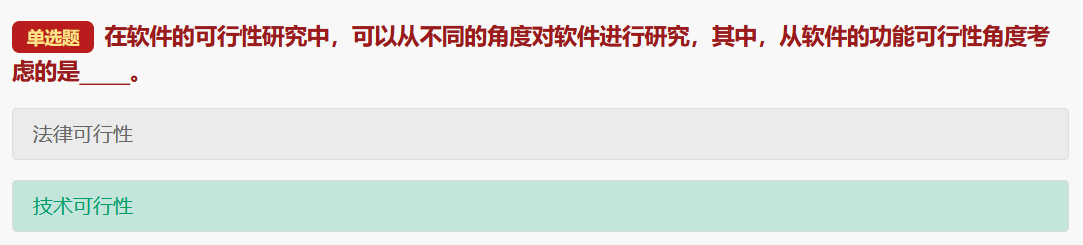


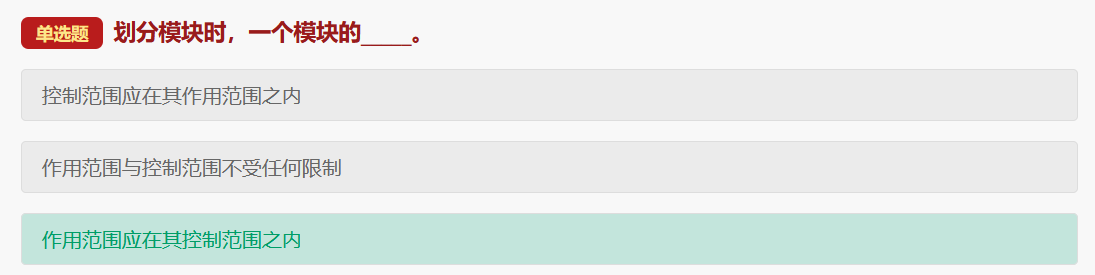






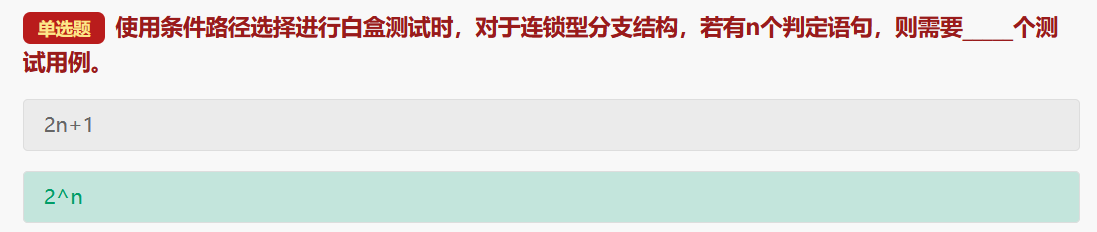




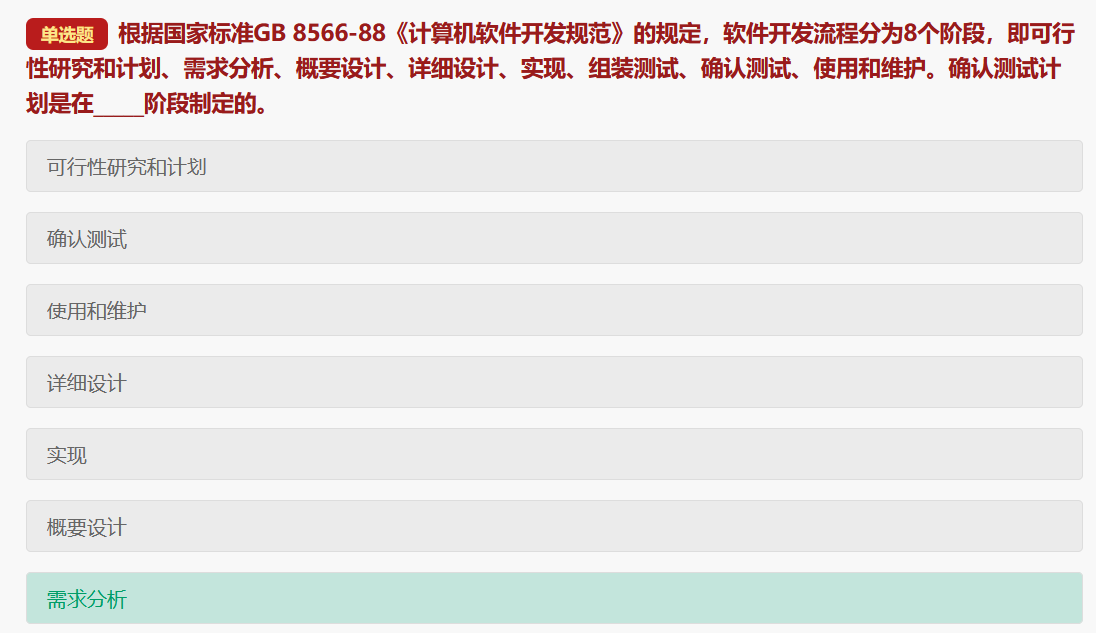


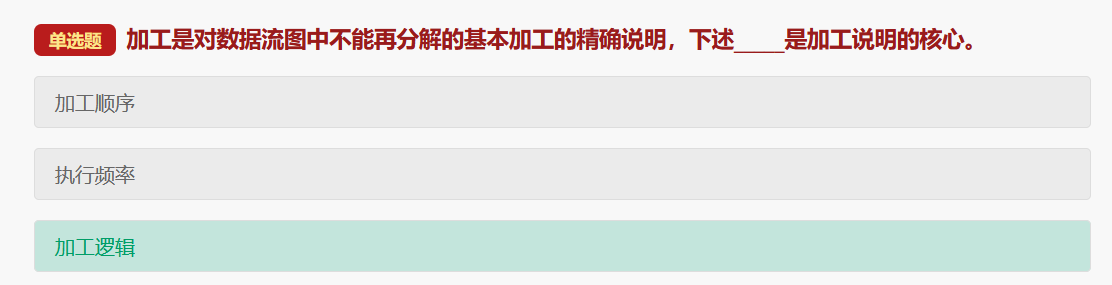


UML不是一种方法

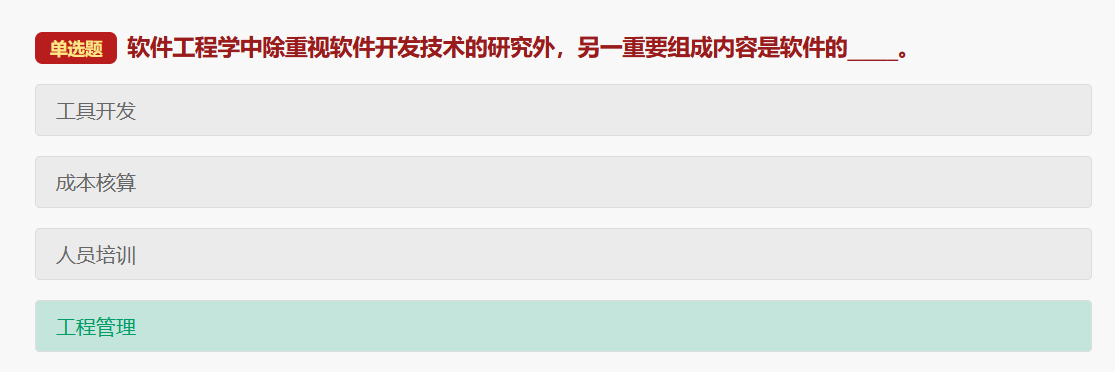








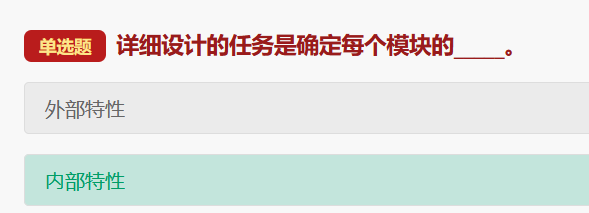


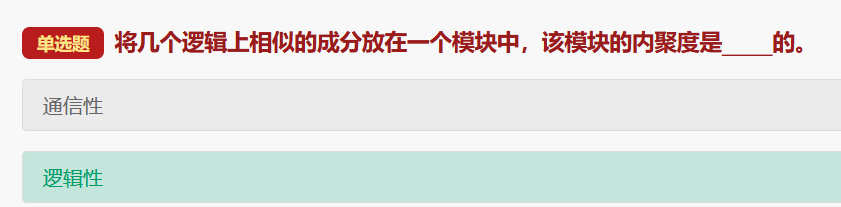


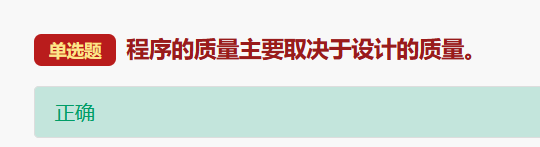


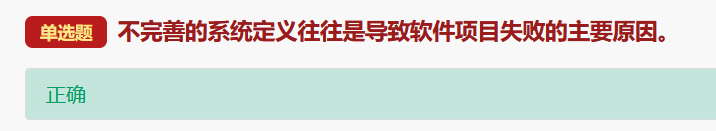


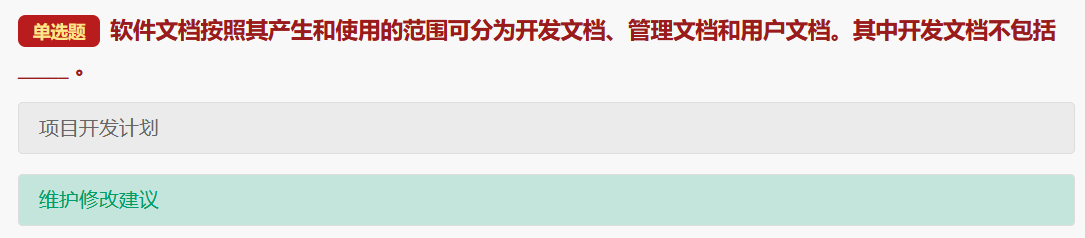


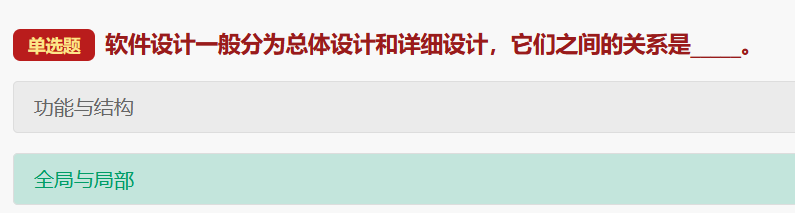






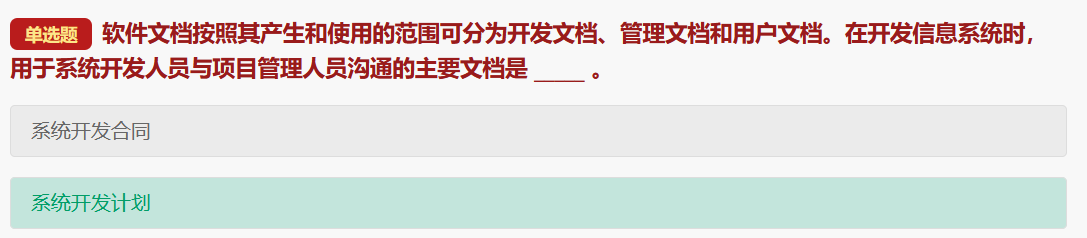




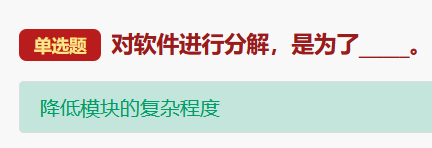


C

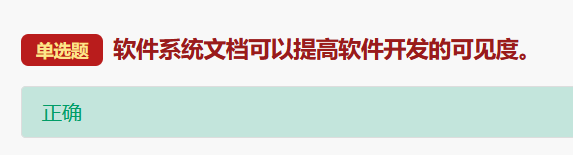


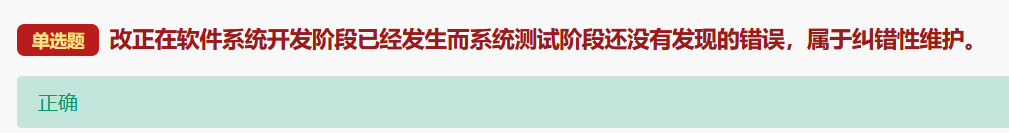




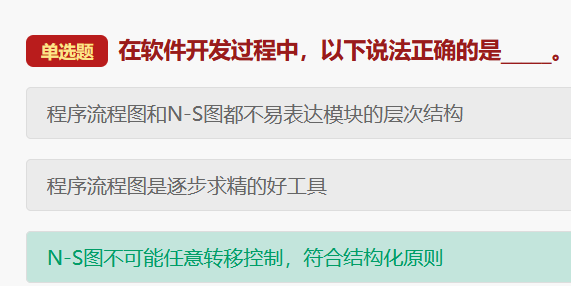


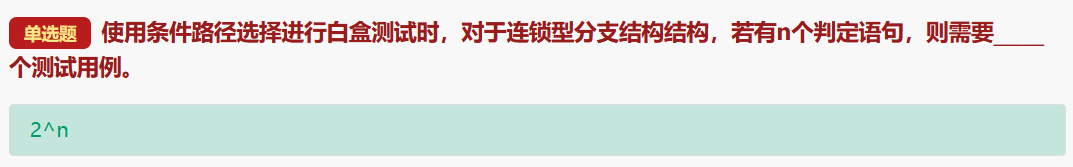




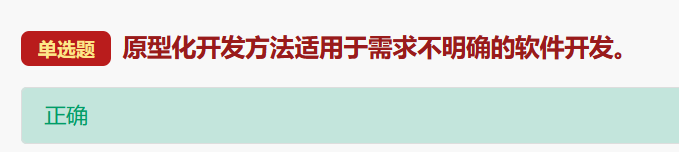




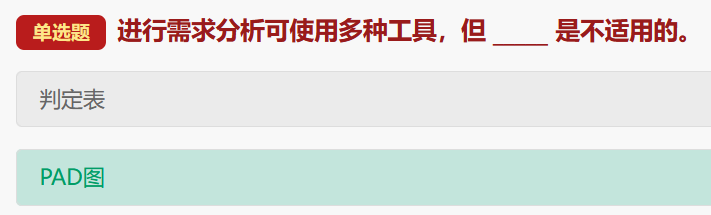


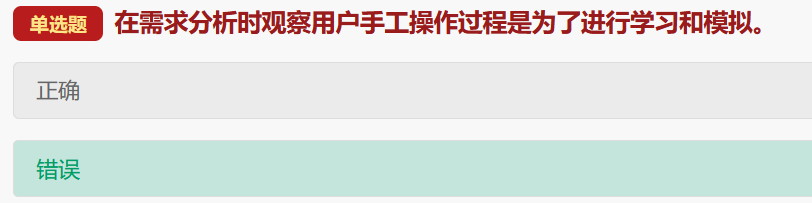


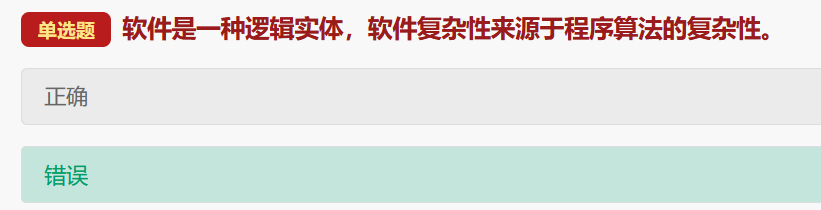




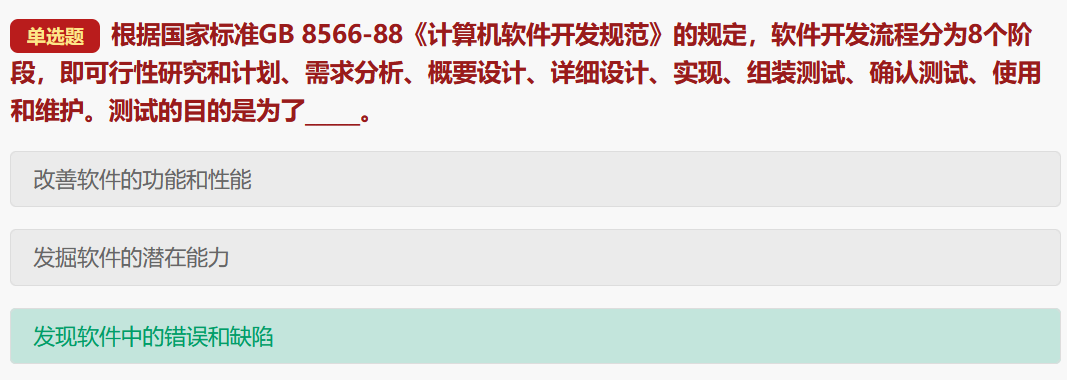






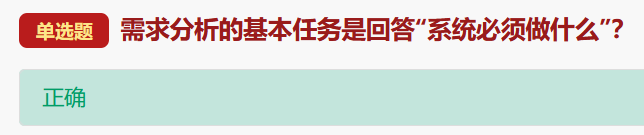




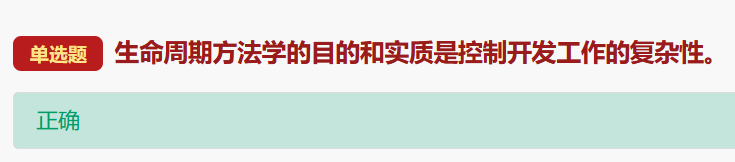




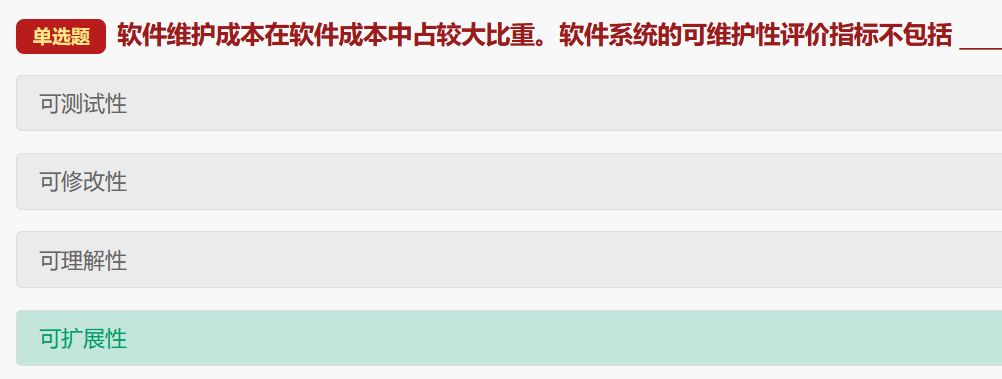




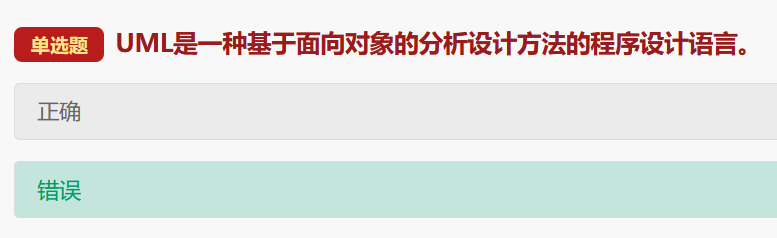


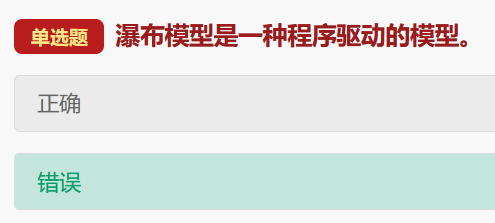




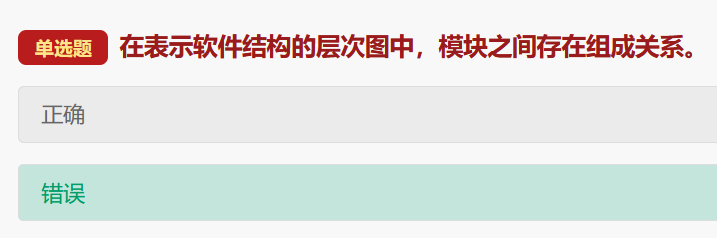


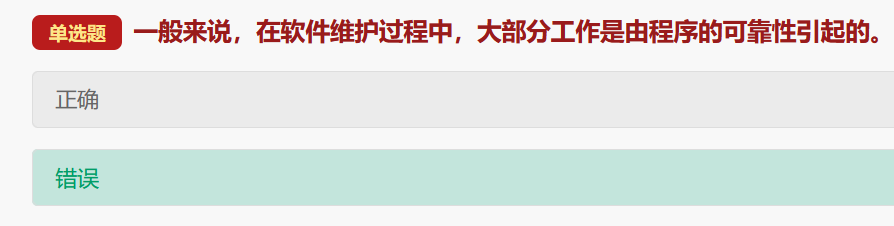






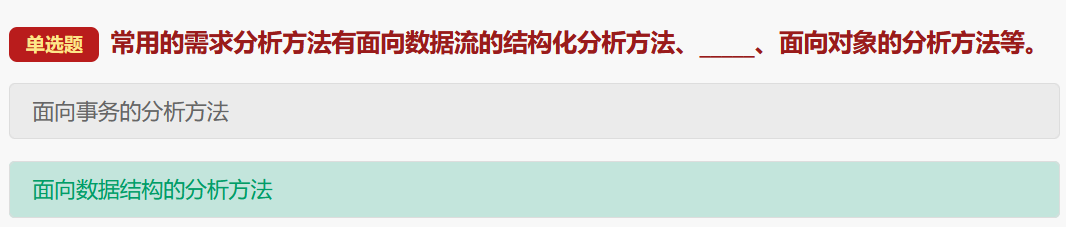
瀑布模型是一个文档驱动模型。

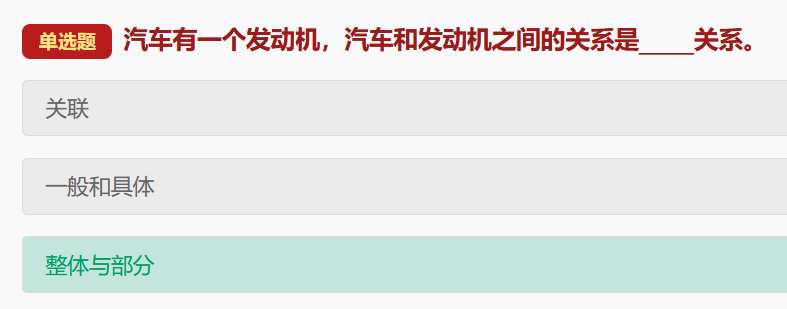




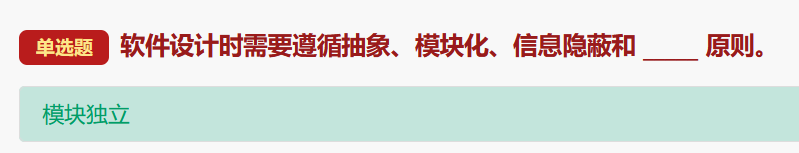
是由于：用户的需求改变 引起的。



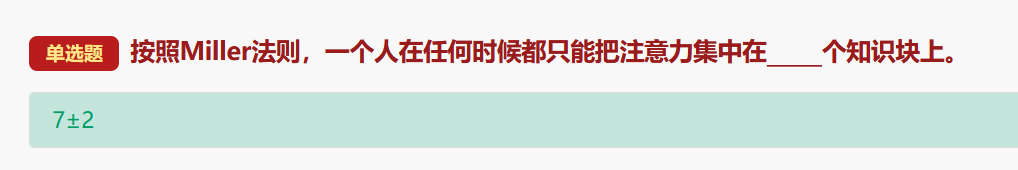


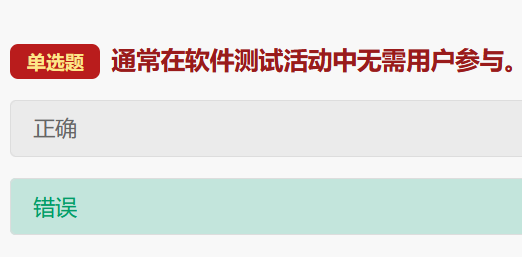




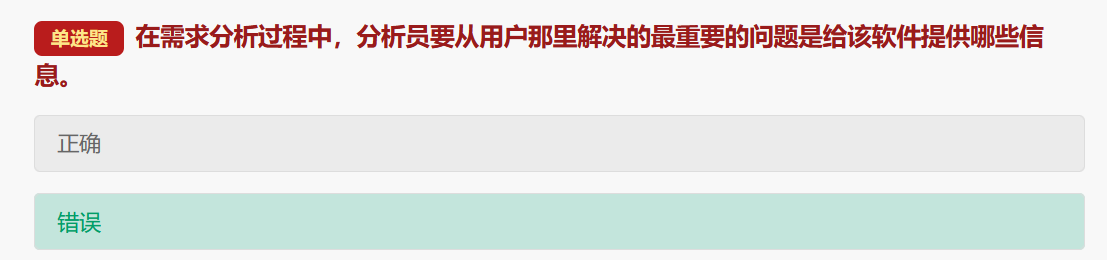






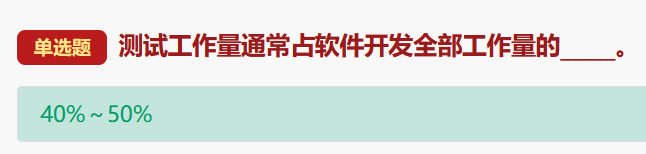


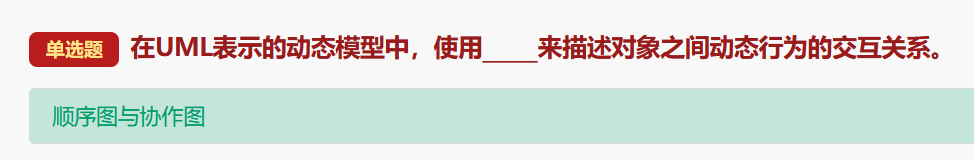










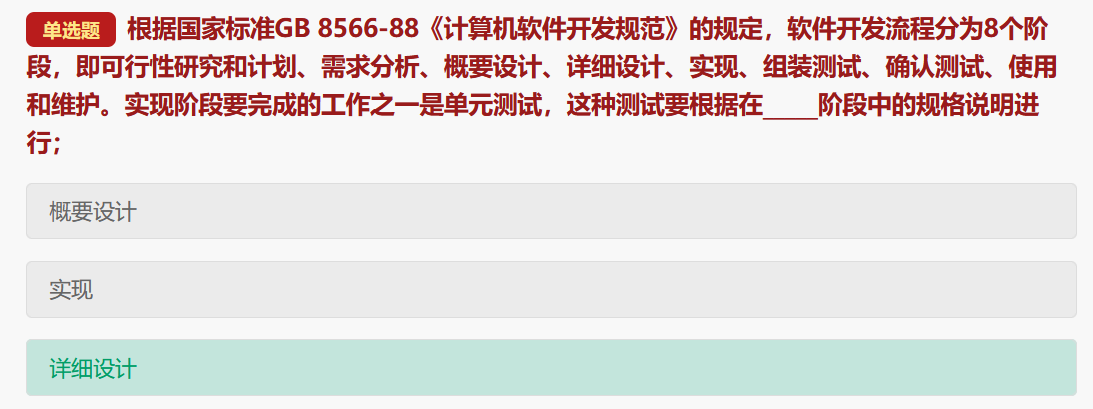


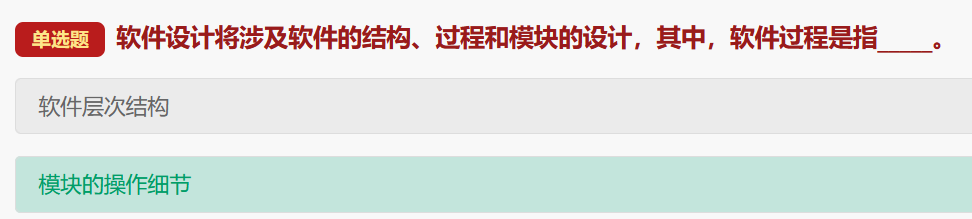


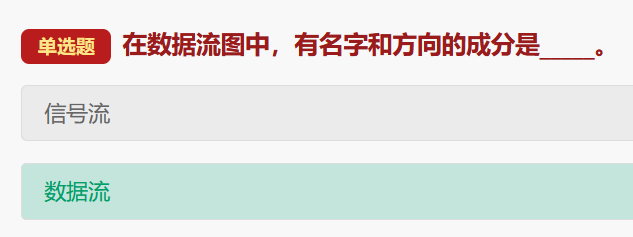


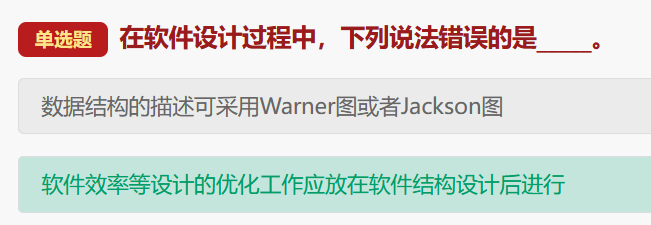
按模块内聚程度从低到高排序为：偶然内聚、逻辑内聚、时间内聚、通信内聚、顺序内聚及功能内聚。

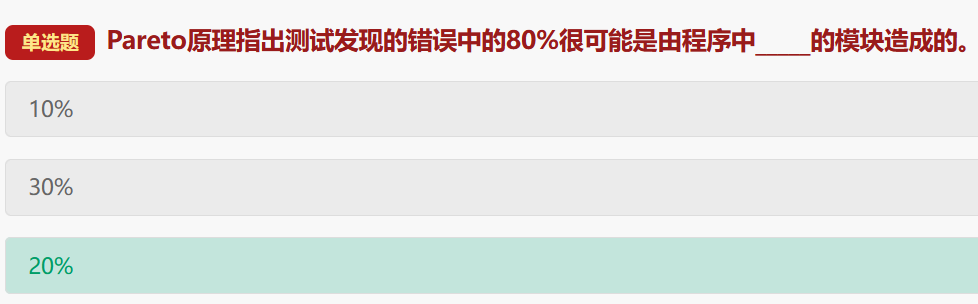




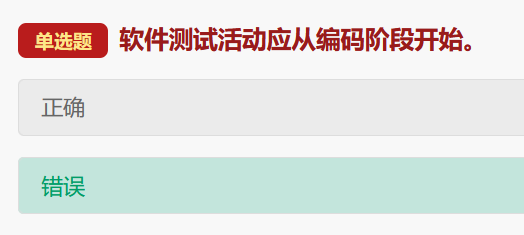


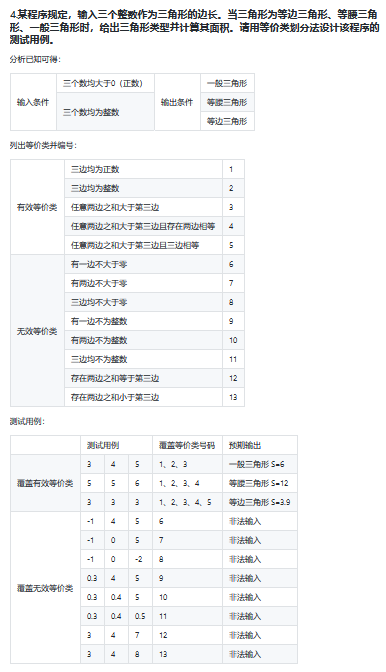












什么是软件生命周期模型？试比较瀑布模型、快速原型模型、增量模型和螺旋模型的优缺点，说明每种模型的适用范围。

    1）软件生命周期由软件定义、软件开发和运行维护3个时期组成，每个时期又进一步划分成若干个阶段。通常使用生命周期模型简洁的描述软件过程，它规定了把生命周期划分成哪些阶段及各个阶段的执行顺序，因此也称过程模型。

    2）四种模型的比较如下：

·瀑布模型

      优点：

           为项目提供了按阶段划分的检查点；

            当前一阶段完成后，只需关注后续阶段；

            可在迭代模型中应用。

      缺点：

            难以在开发初期指明用户全部需求；

            需求确定后，经过一段时间才得到软件最初版本；

            完全依赖规格说明，产品往往反映用户的初始需求而非最终需求。

      适用范围：

            需求明确且几乎不变的中小型项目。

·快速原型模型

      优点：

            满足用户需求程度高、用户的参与面广、返工现象少，减少由于软件需求不明确带来的开发风险。

      缺点：

            所选用的开发技术和工具不一定符合主流的发展；

            快速建立起来的系统结构加上连续的修改可能导致产品质量低下；

            使用前提是要有一个展示性的产品原型，一定程度上可能会限制开发人员的创新。

      适用范围：

            预先不能确切定义需求的软件系统的小型项目。

·增量模型

      优点：

            短期内可以交付满足部分用户需求的功能产品；

            可维护性较好；

            开始第一构件前，已经完成需求说明；

            模块化待开发系统，分批次提交产品，使用户及时了解并适应软件项目的进展；

            降低了开发风险一个错误不会影响到整个系统;

            开发顺序灵活，可先完成需求稳定的核心组件。

      缺点：

            各构件并入已有体系结构中时，需不破坏原有部分，所以应具备开放式的体系结构；

            易退化为边做边改模型，而使软件过程的控制失去整体性；

            若增量包间存在相交的情况且未很好处理，则必须做全盘系统分析。

      适用范围：

            需求经常改变的中型开发项目。

·螺旋模型

      优点：

            集成了上述三种模型的优点，需求分析与软件实现紧密联系相互依赖；

            有利于已有软件的重用和把软件质量作为开发的一个重要目标；

            减少了过多测试（浪费资金）或测试不足（产品故障多）；

            维护只是模型的另一个周期，与开发间没有本质区别。

      缺点：

            需具有相当丰富的风险评估经验和知识，如未能及时标识风险，势必造成重大损失；

            过多的迭代次数会增加开发成本，延迟提交时间。

      适用范围：

            内部开发的大规模软件项目。