软件开发模型之瀑布模型、快速原型模型、增量模型、螺旋模型

软件开发模型：

软件开发模型(Software Development Model)是指软件开发全部过程、活动和任务的结构框架。软件开发包括需求、设计、编码和测试等阶段，有时也包括维护阶段。 软件开发模型能清晰、直观地表达软件开发全过程，明确规定了要完成的主要活动和任务，用来作为软件项目工作的基础。对于不同的软件系统，可以采用不同的开发方法、使用不同的程序设计语言以及各种不同技能的人员参与工作、运用不同的管理方法和手段等，以及允许采用不同的软件工具和不同的软件工程环境。

瀑布模型：

**1、基本概念：**瀑布模型是将软件生存周期的各项活动规定为按固定顺序而连接的若干阶段工作，形如瀑布流水，最终得到软件产品。

**2、核心思想：**采用结构化的分析与设计方法将功能的实现与设计分开，便于分工合作。即将软件生命周期划分为可行性分析、需求分析、概要设计、详细设计、编码实现、测试与维护，六个基本阶段，并规定了他们自上而下，相互衔接的固定次序，如同瀑布流水，逐级下落。

**3、实施过程：**从上一项活动接收该项活动的工作对象作为输入，利用这一输入实施该项活动应完成的内容给出该项活动的工作成果，并作为输出传给下一项活动。同时评审该项活动的实施，若确认，则继续下一项活动；否则返回前面，甚至更前面的活动。

**4、优点：**能够逐步稳定的使项目向前发展。

**5、缺点：**只有在项目完全结束后才能看到软件产品，不能在短期内向用户呈现一个基本软件模型；另外，瀑布模型这种固定次序的开发模式可能会浪费太多不必要的时间。

快速原型模型：

**1、基本概念：**快速原型模型需要迅速建造一个可以运行的软件原型 ，以便理解和澄清问题，使开发人员与用户达成共识，最终在确定的客户需求基础上开发客户满意的软件产品。

**2、核心思想：**在开发真实系统之前，构造一个原型，在该原型的基础上，逐渐完成整个系统的开发工作。

**3、实施过程：**第一步是建造一个快速原型，实现客户或未来的用户与系统的交互，用户或客户对原型进行评价，进一步细化待开发软件的需求。通过逐步调整原型使其满足客户的要求，开发人员可以确定客户的真正需求是什么;第二步则在第一步的基础上开发客户满意的软件产品。

**4、优点：**能够有效减少由于软件需求不明确带来的开发风险。

**5、缺点：**快速建立起来的系统结构可能会在后续的持续更改中导致软件质量低下。

增量模型：

**1、基本概念：**增量模型就是先实现软件的主要核心功能满足用户需求，但一些补充性功能还未发布，通过后期不断完善迭代，最终实现一个完善的软件产品。

**2、核心思想：**本质上是迭代的，但与原型实现不一样的是其强调每一个增量均发布一个可操作产品。早期的增量是最终产品的“可拆卸”版本，但提供了为用户服务的功能，并且为用户提供了评估的平台。即，瀑布模型的不断重复迭代过程。

**3、实现过程：** 当使用增量模型时，第1个增量往往是核心的产品，即第1个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。增量模型强调每一个增量均发布一个可操作的产品。

**4、优点：**能够快速推出软件产品，满足客户需求，对客户有一定的镇定作用，同时不用一下子投入太多资源。灵活性比较高。

**5、缺点：**由于软件其他构件是后期加入的，容易导致由于构件之间的不稳定从而使系统崩溃。

螺旋模型：

**1、基本概念：**是一种演化软件开发过程模型，它兼顾了快速原型的迭代的特征以及瀑布模型的系统化与严格监控。螺旋模型最大的特点在于引入了其他模型不具备的风险分析，使软件在无法排除重大风险时有机会停止，以减小损失。

**2、核心思想：**该模型是快速原型法，以进化的开发方式为中心，在每个项目阶段使用瀑布模型法。这种模型的每一个周期都包括需求定义、风险分析、工程实现和评审4个阶段，由这4个阶段进行迭代。软件开发过程每迭代一次，软件开发又前进一个层次。

**3、实现过程：**螺旋模型基本做法是在"瀑布模型"的每一个开发阶段前引入一个非常严格的风险识别、风险分析和风险控制，它把软件项目分解成一个个小项目。每个小项目都标识一个或多个主要风险，直到所有的主要风险因素都被确定。

**4、优点：**能够使客户每个阶段都参与开发，设计上更加灵活，能始终了解客户新的需求。

**5、缺点：**建设周期长，而软件技术发展比较快，所以经常出现软件开发完毕后，和当前的技术水平有了较大的差距，无法满足当前用户需求。

**总结：**总之，没有完美的开发模型，每种开发模型都有其优点，不同的开发模型适用于不同的软件项目，有时候需要集中模型同时使用等。