

软件开发中各种开发模型（方式）的优缺点

产品名称	软件开发中各种开发模型（方式）的优缺点
公司名称	河南聚匠信息科技有限公司
价格	.00/个
规格参数	
公司地址	金水区广电南路与经三路交叉口金印现代城6号楼2007
联系电话	19937650370

产品详情

可能新手或初级工程师不会在乎什么开发模型，管他三七二十一，直接开干。

但有经验的工程师都会考虑的比较周全，计划、需求、设计等各个环节考虑清楚才开始编码。下面来说说软件开发中常见的开发模型。

瀑布模型

该模型是由上至下一次性完成整个项目的开发方式。该模型一共分为6个阶段，如图所示：

在瀑布模型的开发过程中需要严格的按照这条线执行，只有完成当前阶段之后才能够进行下一阶段的开发任务。

优点

- 1、该模型划分出了每个阶段的检查点，当一个阶段开发完成之后，开发人员的精力可以全部的投入下个阶段，有利于提高开发效率，便于项目的管理。
- 2、比较适用于前期的软件开发与小型软件系统的开发中。

缺点

- 1、无法评估项目进度。因为不知道哪个阶段会造成项目的延期
- 2、无法适应用户的需求变更，只能等到项目完成后，用户才能够看到项目结果

快速原型模型

快速原型模型与瀑布模型相反，项目初期根据用户的需求快速构建一个可以运行的系统原型，之后向用

户展示，由用户进行审核，提出意见，然后逐步丰富项目需求。当需求真正确定后，才正式进行项目开发。模型如图所示：

解决需求不明确带来的风险，适用于不能提前确定项目需求的项目

不利于开发人员对产品进行扩展

迭代模型

迭代模型又被称作为增量模型或演化模型，它将一个完整的软件拆分成不同的组件，之后对每个组件进行逐步的开发测试，每当完成一个组件就会向客户进行展示，让客户确认该组件功能与性能是否达到要求，最终确定无误，将组件集成到软件体系结构中。整个开发工作被分为为一系列短期、简单的小项目，称为一系列迭代，每一个迭代都需要经过需求分析、软件设计、编码、测试这几项过程，其开发过程如图所示：

- 1、低以个可交付版本的软件所需的成本与时间较小
- 2、能够适应客户的需求变更，当需求变化时，只需要修改某一个组件即可。
- 1、如果对用户需求的变更没有整体的规划，可能会变化为"边做边开发"的模式。
- 2、最终集成各个组件时，可能会出现集成失败的风险。

喷泉模型

该模型主要采用面向对象技术。当客户需求基本类似时，在开发过程中可以采用面向对象的开发方式，将相同的模块全部封装起来，以便于下次功能开发时使用。模型如图所示：

支持软件重用，并且开发过程无间隙性，分析、设计编码无明显边界，可交叉迭代进行。使软件在无法排除重大风险时有机会停止，以减小损失。

- 1、由于喷泉模型在各个阶段是重叠的，即每个对象都有分析、设计和编码阶段，所以需要大量开发人员。
- 2、大量开发人员不利于项目的管理。
- 3、该模型需要严格管理文档，会增加审核的难度增大。

螺旋模型

螺旋模型融合了瀑布模型，快速原型模型，该模型最大的特点就是引入了其他模型所没有的风险分析。

螺旋模型将开发过程都分为几个螺旋周期，每个螺旋周期大致和瀑布模型相符合，在每个周期开始之前都会进行风险分析。在每个迭代阶段构建原型是螺旋模型用以减小风险的途径。模型如图所示：

该模型共有四个象限，每个象限的含义如下：

- 1、制订计划:确定软件目标，制订实施方案，并且列出项目开发的限制条件。
- 2、风险分析:评价所制订的实施方案，识别风险并消除风险。

3、实施工程:开发产品并进行验证。

4、客户评估:客户对产品进行审核评估，提出修正建议，制订下一步计划。

螺旋模型强调风险分析，对每个演化层出现的风险都有所了解，继而做出应有反应。因此特别适合用于庞大、复杂并且具有高风险的系统。螺旋模型支持用户需求的动态变化有助于提高产品的适应能力。

过多的迭代次数会增加开发成本，延迟提交时间。

敏捷模型

在现代社会的开发中，由于业务会经常快速的变化，因此会导致在软件开发之前经常是无法得到详细完整的开发需求，没有完整的开发需求，传统的软件开发模型也就无法适用。

敏捷开发模型的提出就是为了解决该问题。该模型以客户的需求为核心，采用迭代，循序渐进的方法进行开发。

软件项目在构建初期会被拆分为多个相互联系而又独立运行的子项目，然后迭代完成各个子项目。开发过程中，各个子项目都要经过开发测试。当客户有需求变更时，敏捷模型能够迅速地对某个子项目做出修改以满足客户的需求。在这个过程中，软件一直处于可使用状态。

该模型更重视人在软件开发中的作用。软件开发过程中，各个部门需要紧密的合作沟通，为适应软件需求的频繁改变，客户可以全程参与到开发过程中。

敏捷开发模型的价值与原则

1、个体和交互重于过程和工具

2、可用软件重于完备文档

3、客户协作重于合同谈判

4、响应变化重于遵循计划

1、用户很快可以看到一个基线架构版的产品

2、敏捷注重市场快速反应能力，客户前期满意度高。

1、注重人员的沟通

2、忽略文档的重要性

3、如果项目人员流动太大，会增加项目维护难度

4、软件之前版本的可重现性、可回溯性较低

5、对于较大的项目,人员越多,面对面的有效沟通越困难。因此，该模型适用于小型项目的开发。