

## APP开发的重要细节

产品名称	APP开发的重要细节
公司名称	东莞市数云网络科技有限公司
价格	.00/个
规格参数	
公司地址	东莞市
联系电话	18665158422 18665158422

## 产品详情

其实现在许多产品实际都是使用的“边做边改”模型来开发的，特别是很多小公司产品周期压缩的太短。在这种模型中，既没有规格说明，也没有经过设计，软件随着客户的需要一次又一次地不断被修改。

在这个模型中，开发人员拿到项目立即根据需求编写程序，调试通过后生成软件的个版本。在提供给用户使用后，如果程序出现错误，或者用户提出新的要求，开发人员重新修改代码，直到用户和测试等等满意为止。

这是一种类似作坊的开发方式，边做边改模型的优点毫无疑问就是前期出成效快。

对编写逻辑不需要太严谨的小程序来说还可以对付得过去，但这种方法对任何规模的开发来说都是不能令人满意的，其主要问题在于：

缺少规划和设计环节，软件的结构随着不断的修改越来越糟，导致无法继续修改；

忽略需求环节，给软件开发带来很大的风险；

没有考虑测试和程序的可维护性，也没有任何文档，软件的维护十分困难。

## 2. 瀑布模型 ( Waterfall Model )

瀑布模型是一种比较老旧的软件开发模型，1970年温斯顿·罗伊斯提出了“瀑布模型”，直到80年代都还是一直被广泛采用的模型。

瀑布模型将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。

在瀑布模型中，软件开发的各项活动严格按照线性方式进行，当前活动接受上一项活动的工作结果，实施完成所需的工作内容。当前活动的工作结果需要进行验证，如验证通过，则该结果作为下一项活动的输入，继续进行下一项活动，否则返回修改。

瀑布模型优点是严格遵循预先计划的步骤顺序进行，一切按部就班比较严谨。

瀑布模型强调文档的作用，并要求每个阶段都要仔细验证。但是，这种模型的线性过程太理想化，已不再适合现代的软件开发模式，几乎被业界抛弃，其主要问题在于：

各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量；

由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发的风险；

早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果。

各个软件生命周期衔接花费时间较长，团队人员交流成本大。

瀑布式方法在需求不明并且在项目进行过程中可能变化的情况下基本是不可行的。

### 3. 迭代模型 (stagewise model)

迭代模型 (也被称作迭代增量式开发或迭代进化式开发) 是一种与传统的瀑布式开发相反的软件开发过程, 它弥补了传统开发方式中的一些弱点, 具有更高的成功率和生产率。

在迭代式开发方法中, 整个开发工作被组织为一系列的短小的、固定长度 (如3周) 的小项目, 被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试。采用这种方法, 开发工作可以在需求被完整地确定之前启动, 并在一次迭代中完成系统的一部分功能或业务逻辑的开发工作。再通过客户的反馈来细化需求, 并开始新一轮的迭代。

教学中, 对迭代和版本的区别, 可理解如下:

迭代一般指某版本的生产过程, 包括从需求分析到测试完成;

版本一般指某阶段软件开发的结果, 一个可交付使用的产品。

与传统的瀑布模型相比较, 迭代过程具有以下优点:

降低了在一个增量上的开支风险。如果开发人员重复某个迭代, 那么损失只是这一个开发有误的迭代的花费。

降低了产品无法按照既定进度进入市场的风险。通过在开发早期就确定风险, 可以尽早来解决而不至于在开发后期匆匆忙忙。

加快了整个开发工作的进度。因为开发人员清楚问题的焦点所在, 他们的工作会更有效率。

由于用户的需求并不能在一开始就作出完全的界定, 它们通常是在后续阶段中不断细化的。因此, 迭代

过程这种模式使适应需求的变化会更容易些。因此复用性更高

#### 4. 快速原型模型 ( Rapid Prototype Model )

快速原型模型的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；第二步则在步的基础上开发客户满意的软件产品。

显然，快速原型方法可以克服瀑布模型的缺点，减少由于软件需求不明确带来的开发风险，具有显著的效果。

快速原型的关键在于尽可能快速地建造出软件原型，一旦确定了客户的真正需求，所建造的原型将被丢弃。因此，原型系统的内部结构并不重要，重要的是必须迅速建立原型，随之迅速修改原型，以反映客户的需求。

快速原型模型有点整合“边做边改”与“瀑布模型”优点的意味。

#### 5、增量模型 ( Incremental Model )

与建造大厦相同，软件也是一步一步建造起来的。在增量模型中，软件被作为一系列的增量构件来设计、实现、集成和测试，每一个构件是由多种相互作用的模块所形成的提供特定功能的代码片段构成。

增量模型在各个阶段并不交付一个可运行的完整产品，而是交付满足客户需求的一个子集的可运行产品。整个产品被分解成若干个构件，开发人员逐个构件地交付产品，这样做的好处是软件开发可以较好地适应变化，客户可以不断地看到所开发的软件，从而降低开发风险。但是，增量模型也存在以下缺陷：

由于各个构件是逐渐并入已有的软件体系结构中的，所以加入构件必须不破坏已构造好的系统部分，这

需要软件具备开放式的体系结构。

在开发过程中，需求的变化是不可避免的。增量模型的灵活性可以使其适应这种变化的能力大大优于瀑布模型和快速原型模型，但也很容易退化为边做边改模型，从而是软件过程的控制失去整体性。

在使用增量模型时，个增量往往是实现基本需求的核心产品。核心产品交付用户使用后，经过评价形成下一个增量的开发计划，它包括对核心产品的修改和一些新功能的发布。这个过程在每个增量发布后不断重复，直到产生终的完善产品。

例如，使用增量模型开发字处理软件。可以考虑，个增量发布基本的文件管理、编辑和文档生成功能，第二个增量发布更加完善的编辑和文档生成功能，第三个增量实现拼写和文法检查功能，第四个增量完成的页面布局功能。

## 6. 螺旋模型 (Spiral Model)

1988年，巴利·玻姆(Barry Boehm)正式发表了软件系统开发的“螺旋模型”，它将瀑布模型和快速原型模型结合起来，强调了其他模型所忽视的风险分析，特别适合于大型复杂的系统。

螺旋模型沿着螺线进行若干次迭代，图中的四个象限代表了以下活动：

制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件；

风险分析：分析评估所选方案，考虑如何识别和消除风险；

实施工程：实施软件开发和验证；

客户评估：评价开发工作，提出修正建议，制定下一步计划。

螺旋模型由风险驱动，强调可选方案和约束条件从而支持软件的重用，有助于将软件质量作为特殊目标融入产品开发之中。但是，螺旋模型也有一定的限制条件，具体如下：

螺旋模型强调风险分析，但要求许多客户接受和相信这种分析，并做出相关反应是不容易的，因此，这种模型往往适应于内部的大规模软件开发。

如果执行风险分析将大大影响项目的利润，那么进行风险分析毫无意义，因此，螺旋模型只适合于大规模软件项目。

软件开发人员应该擅长寻找可能的风险，准确地分析风险，否则将会带来更大的风险

一个阶段首先是确定该阶段的目标，完成这些目标的选择方案及其约束条件，然后从风险角度分析方案的开发策略，努力排除各种潜在的风险，有时需要通过建造原型来完成。如果某些风险不能排除，该方案立即终止，否则启动下一个开发步骤。后，评价该阶段的结果，并设计下一个阶段。

## 7. 敏捷软件开发 (Agile development)

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法。在敏捷开发中，软件项目的构建被切分成多个子项目，各个子项目的成果都经过测试，具备集成和可运行的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

敏捷开发小组主要的工作方式可以归纳为：作为一个整体工作；按短迭代周期工作；

每次迭代交付一些成果，关注业务优先级，检查与调整。

敏捷软件开发要注意项目规模，规模增长，团队交流成本就上去了，因此敏捷软件开发暂时适合不是特别大的团队开发，比较适合一个组的团队使用。

## 8. 演化模型 ( evolutionary model )

主要针对事先不能完整定义需求的软件开发。用户可以给出待开发系统的核心需求，并且当看到核心需求实现后，能够有效地提出反馈，以支持系统的终设计和实现。软件开发人员根据用户的需求，首先开发核心系统。当该核心系统投入运行后，用户试用之，完成他们的工作，并提出精化系统、增强系统能力的需要。软件开发人员根据用户的反馈，实施开发的迭代过程。迭代过程均由需求、设计、编码、测试、集成等阶段组成，为整个系统增加一个可定义的、可管理的子集。

在开发模式上采取分批循环开发的办法，每循环开发一部分的功能，它们成为这个产品的原型的新增功能。于是，设计就不断地演化出新的系统。实际上，这个模型可看作是重复执行的多个“瀑布模型”。

“演化模型”要求开发人员有能力把项目的产品需求分解为不同组，以便分批循环开发。这种分组并不是随意性的，而是要根据功能的重要性及对总体设计的基础结构的影响而作出判断。有经验指出，每个开发循环以六周到八周为适当的长度。

## 9. 喷泉模型 ( fountain model, (面向对象的生存期模型, 面向对象 ( Object Oriented,OO ) 模型 ) )

喷泉模型与传统的结构化生存期比较，具有更多的增量和迭代性质，生存期的各个阶段可以相互重叠和多次反复，而且在项目的整个生存期中还可以嵌入子生存期。就像水喷上去又可以落下来，可以落在中间，也可以落在底部。

## 10. 智能模型 ( 四代技术 ( 4GL ) )

智能模型拥有一组工具 ( 如数据查询、报表生成、数据处理、屏幕定义、代码生成、高层图形功能及电子表格等 ) ，每个工具都能使开发人员在高层次上定义软件的某些特性，并把开发人员定义的这些软件自动地生成为源代码。这种方法需要四代语言 ( 4GL ) 的支持。4GL不同于三代语言，其主要特征是用户界面极端友好，即使没有受过训练的非程序员，也能用它编写程序；它是一种声明式、交互式和非过程

性编程语言。4GL还具有高效的程序代码、智能缺省假设、完备的数据库和应用程序生成器。目前市场上流行的4GL（如Foxpro等）都不同程度地具有上述特征。但4GL目前主要限于事务信息系统的中、小型应用程序的开发。

#### 11. 混合模型（hybrid model）

过程开发模型又叫混合模型（hybrid model），或元模型（meta-model），把几种不同模型组合成一种混合模型，它允许一个项目能沿着有效的路径发展，这就是过程开发模型（或混合模型）。实际上，一些软件开发单位都是使用几种不同的开发方法组成他们自己的混合模型。