

邯郸软件开发公司|邯郸软件公司|软件定制开发|软件开发

产品名称	邯郸软件开发公司 邯郸软件公司 软件定制开发 软件开发
公司名称	河北创客情创业服务有限公司
价格	15000.00/套
规格参数	
公司地址	河北省邯郸市邯山区光明南大街城市新秀商务大厦18层
联系电话	0310-3334555 18303030203

产品详情

一、基本定义

软件开发是根据用户要求建造出软件系统或者系统中软件部分的一个产品开发的过程。软件开发是一项包括需求获取、开发规划、需求分析和设计、编程实现、软件测试、版本控制的系统工程。换句话说，软件开发就是一系列终构建出软件产品的活动。

二、阶段划分

1、计划

对所要解决的问题进行总体定义，包括了解用户的要求及现实环境，从技术、经济和社会因素等3个方面研究并论证本软件项目的可行性，编写可行性研究报告，探讨解决问题的方案，并对可供使用的资源（如计算机硬件、系统软件、人力等）成本，可取得的效益和开发进度作出估计，制订完成开发任务的实施计划。

2、分析

软件需求分析就是对开发什么样的软件的一个系统的分析与设想。它是一个对用户的需求进行去粗取精、去伪存真、正确理解，然后把它用软件工程开发语言（形式功能规约，即需求规格说明书）表达出来的过程。本阶段的基本任务是和用户一起确定要解决的问题，建立软件的逻辑模型，编写需求规格说明书文档并终得到用户的认可。需求分析的主要方法有结构化分析方法、数据流程图和数据字典等方法。本阶段的工作是根据需求说明书的要求，设计建立相应的软件系统的体系结构，并将整个系统分解成若干个子系统或模块，定义子系统或模块间的接口关系，对各子系统进行具体设计定义，编写软件概要设计和详细设计说明书，数据库或数据结构设计说明书，组装测试计划。在任何软件或系统开发的初始阶段必须先完全掌握用户需求，以期能将紧随的系统开发过程中哪些功能应该落实、采取何种规格以及设定哪些限制优先加以定位。系统工程师终将据此完成设计方案，在此基础上对随后的程序开发、系统功能和性能的描述及限制作出定义。

3、设计

软件设计可以分为概要设计和详细设计两个阶段。实际上软件设计的主要任务就是将软件分解成模块，然后进行模块设计。模块是指能实现某个功能的数据和程序说明、可执行程序的程序单元。可以是一个函数、过程、子程序、一段带有程序说明的独立的程序和数据，也可以是可组合、可分解和可更换的功能单元。概要设计就是结构设计，其主要目标就是给出软件的模块结构，用软件结构图表示。详细设计的首要任务就是设计模块的程序流程、算法和数据结构，次要任务就是设计数据库，常用方法还是结构化程序设计方法。

4、编码

软件编码是指把软件设计转换成计算机可以接受的程序，即写成以某一程序设计语言表示的“源程序清单”。充分了解软件开发语言、工具的特性和编程风格，有助于开发工具的选择以及保证软件产品的开发质量。

5、测试

软件测试的目的是以较小的代价发现尽可能多的错误。要实现这个目标的关键在于设计一套出色的测试用例（测试数据与功能和预期的输出结果组成了测试用例）。如何才能设计出一套出色的测试用例，关键在于理解测试方法。不同的测试方法有不同的测试用例设计方法。两种常用的测试方法分别是白盒法和黑盒法，白盒法测试对象是源程序，依据的是程序内部的逻辑结构来发现软件的编程错误、结构错误和数据错误。结构错误包括逻辑、数据流、初始化等错误。白盒法用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果。黑盒法依据的是软件的功能或软件行为描述，发现软件的接口、功能和结构错误。其中接口错误包括内部/外部接口、资源管理、集成化以及系统错误。黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。

6、维护

维护是指在已完成对软件的研制（分析、设计、编码和测试）工作并交付使用以后，对软件产品所进行的一些软件工程的活动。即根据软件运行的情况，对软件进行适当修改，以适应新的要求，以及纠正运行中发现的错误。编写软件问题报告、软件修改报告。

三、典型的软件开发模型

软件开发模型是指软件开发全部过程、活动和任务的结构框架。它能清晰、直观地表达软件开发全过程，明确规定要完成的主要活动和任务，用来作为软件项目工作的基础。

1、边做边改模型（Build-and-Fix Model）

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，开发人员就重新修改代码，直到用户满意为止。

这种开发方式对编写几百行的小程序来说还不错，但对任何规模的开发来说都不能令人满意，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)

1970年温斯顿·罗伊斯提出了的“瀑布模型”，直到80年代早期，它一直是唯一被广泛采用的软件开发模型。

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，如果验证通过，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。

3、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。

4、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，从而使软件过程的控制失去整体性。

5、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。

6、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、增强系统能力的需求。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”。

7、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。

8、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。

9、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。