

标准：计算机软件开发规范 GB 8566-88 （已作废）

目的：详细规定计算机软件开发过程各个阶段及没法儿阶段的任务、 实施步骤、实施要求 、完成标志及交付文件。为软件开人员和管理人员提供一系列之有效的准则、方法和规范。

作用：有利于提高开发的控制和管理， 缩短开发时间和减少维护次数， 便于开发和维护人员之间的协作、交流，是软件开发更加有成效。

软件的生存周期： Systems Development Life Cycle (SDLC)

可行性研究与计划

需求分析

概要设计

详细设计

实现

组装测试

确认测试

使用和维护

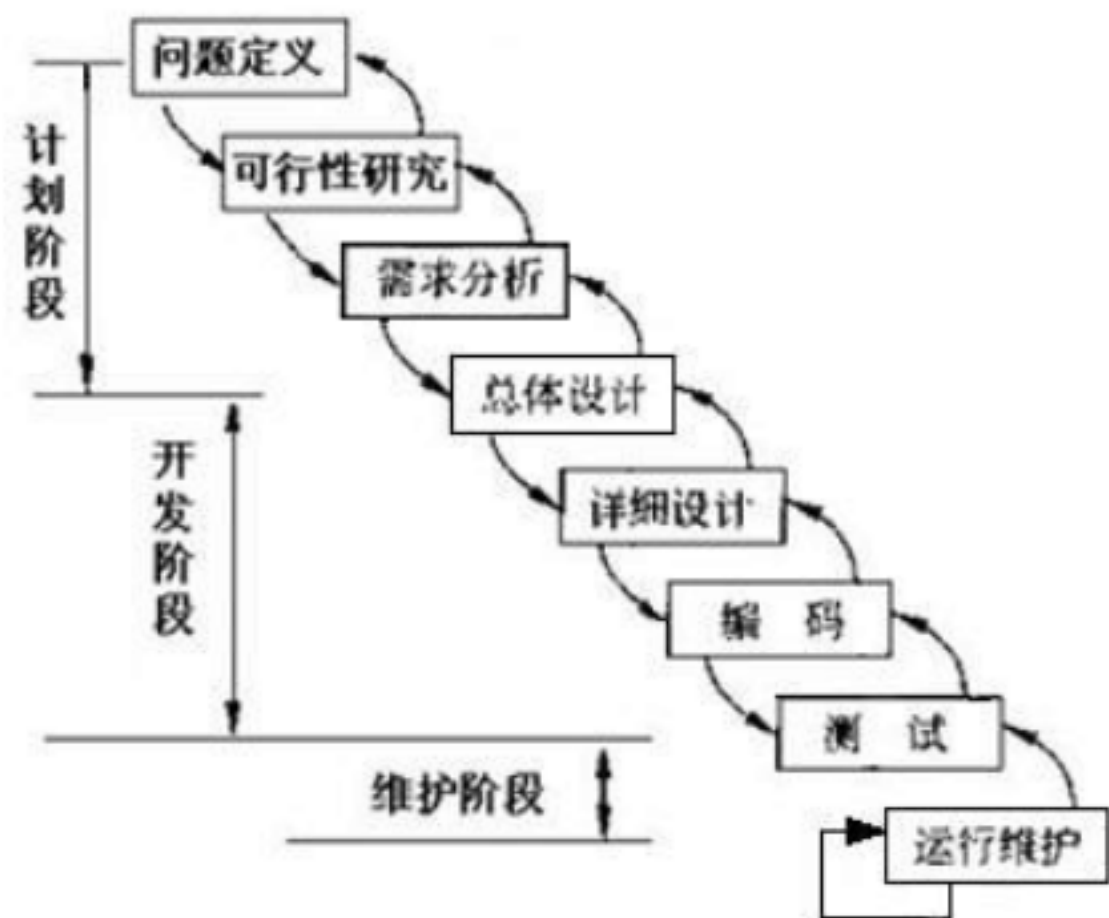
按照人们所习惯的粗分方法把上面 8 个阶段划分为计划、开发和维护 3 个阶段，在概述其他两个阶段的基础上重点介绍软件的开发过程

2. 软件开发方法
求。

模型	特点	适用类型	优点	缺点
瀑布模型：	该模型严格规定各阶段的任务，上一阶段任务输出作为下一阶段工作输入。	此模型适合于用户需求明确、开发技术比较成熟、工程管理严格的场合使用。	为项目提供了按阶段划分的检查点。 当前一阶段完成后，您只需要去关注后续阶段。	由于任务顺序固定，软件研制周期长，前一阶段工作中造成的差错越到后期越大，而且纠正前期错误的代价高。
渐进模型（快速原型模型）：	从一组简单的基本用户需求出发，首先建立一个满足基本要求的原型系统。通过测试和运行原型系统，有用户提出进一步细致的需求，然后修改和完善原型系统，反复进行这个过程直到用户	该模型适合开发初期用户需求不甚明确，相关技术和理论需要不断研究、反复实验以及开发过程需要经常与用户交互的场合，学习或研究类软件的开发常用此法。	由于用户在整个软件开发过程中都直接参与，因此最终的软件产品能够很好地满足用户的需求。	

	满意为止。			
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	<p>1)设计上的灵活性 , 可以在项目的各个阶段进行变更。</p> <p>2)以小的分段来构建大型系统 , 使成本计算变得简单容易。</p> <p>3)客户始终参与每个阶段的开发 , 保证了项目不偏离正确方向以及项目的可控性。</p> <p>4) 随着项目推进 , 客户始终掌握项目的最新信息 , 从而他或她能够和管理层有效地交互。</p> <p>5)客户认可这种公司内部的开发方式带来的良好的沟通和高质量的产品。</p>	对于新近开发 , 需求不明确的情况下 , 适合用螺旋模型进行开发 , 便于风险控制和需求变更 !
喷泉模型 :	<p>特点是各项活动之间没有明显的界限。</p> <p>该开发过程分为 : 分析、设计、实现、确认、维护和演化。</p>	该模型主要用于面向对象软件技术开发项目	该模型软件开发过程与开发者对问题认识 and 理解的深化过程同步。该模型重视软件研发工作的重复与渐进 , 通过相关对象的反复迭代并在迭代中充实扩展 , 实现了开发工作的迭代和无间隙 ,	
软件测试模型 (V 模型)	旨在改进软件开发的效率和效果。V 模型反映出了测试活动与分析设计活动的关系			V 模型存在一定的局限性 , 它仅仅把测试作为在编码之后的一个阶段 , 是针对程序进行的寻找错

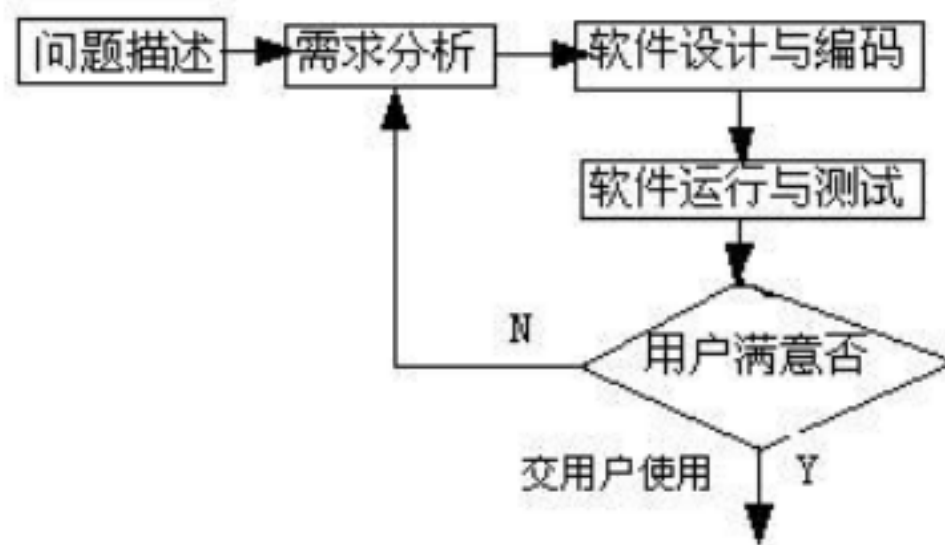
				误的活动，而忽视了测试活动对需求分析、系统设计等活动的验证和确认的功能
W 模型	相对于 V 模型，W 模型增加了软件各开发阶段中应同步进行的验证和确认活动		测试伴随着整个软件开发周期，而且测试的对象不仅仅是程序，需求、设计等同样要测试，也就是说，测试与开发是同步进行的。W 模型有利于尽早地全面的发现问题。例如，需求分析完成后，测试人员就应该参与到对需求的验证和确认活动中，以尽早地找出缺陷所在。同时，对需求的测试也有利于及时了解项目难度和测试风险，及早制定应对措施，这将显著减少总体测试时间，加快项目进度。	但 W 模型也存在局限性。在 W 模型中，需求、设计、编码等活动被视为串行的，同时，测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一个阶段工作。这样就无法支持迭代的开发模型。对于当前软件开发复杂多变的情况，W 模型并不能解除测试管理面临着困惑



瀑布模型

阶段	基本任务	工作结果
问题定义	理解问题	系统目标与范围说明书
可行性研究	理解工作范围	项目计划任务书
需求分析	定义用户需求	需求规格说明书
总体设计	建立软件结构	总体设计说明书
详细设计	模块功能实现	程序规格说明书
编码	编写程序	程序清单
测试	发现错误、调试	软件产品
运行维护	运行和管理	改进的软件产品

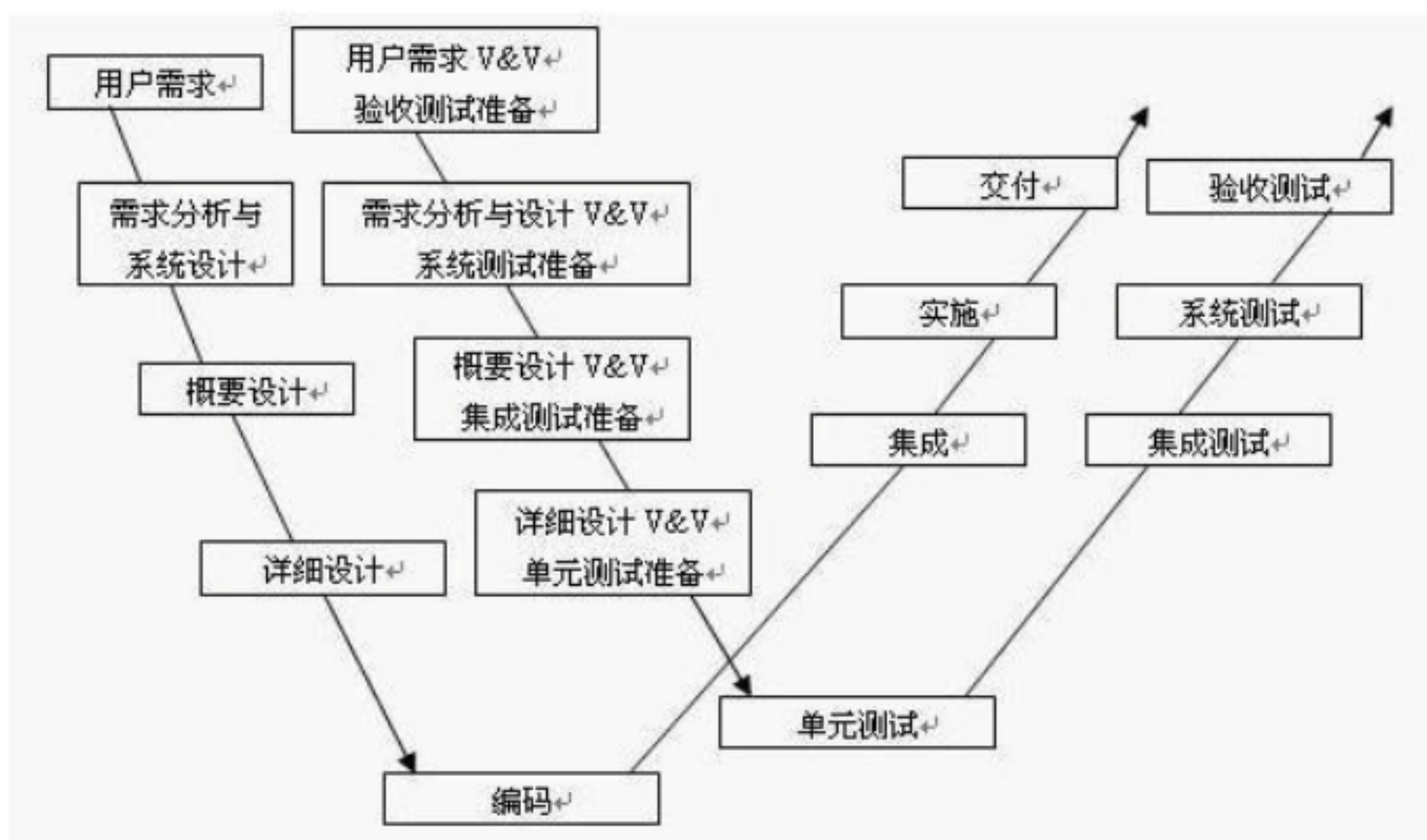
瀑布模型阶段任务



渐进模型



V 模型



双 v 模型

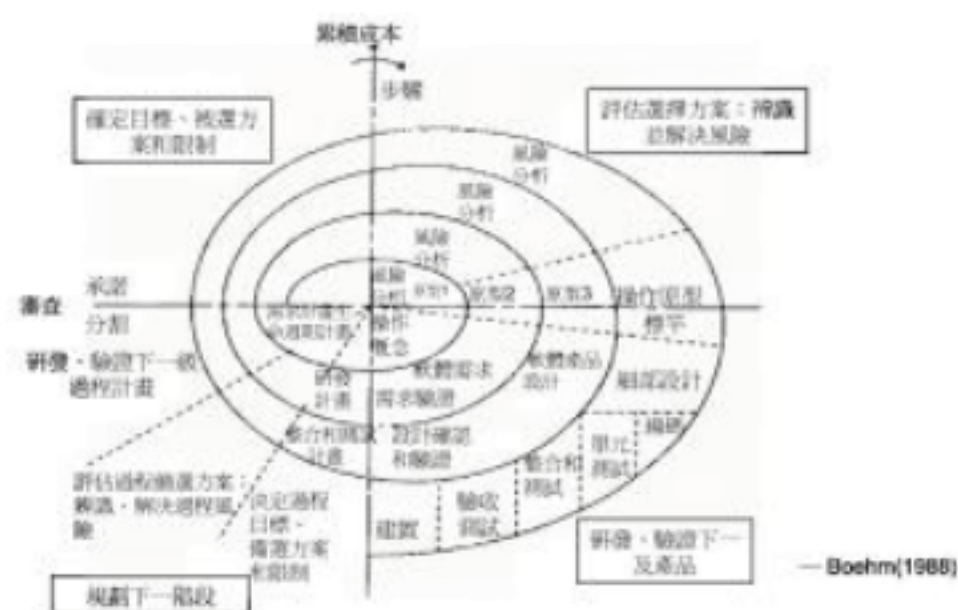


图 5 螺旋模型(The spiral model)

(注：由 R. E. Thayer 和 M. Dorfman 编写的 IEEE 标准《系统地软件需求工程》一书介绍了 Barry Boehm 的“软件需求和开发的螺旋模型”(1990 年版本。)

螺旋模型

快速原型（ Rapid Prototype ）模型：快速原型模型在功能上等价于产品的一个子集。注意，这里说的是功能上。瀑布模型的缺点就在于不够直观，快速原型法就解决了这个问题。一般来说，根据客户的需要在很短的时间内解决用户最迫切需要，完成一个可以演示的产品。这个产品只是实现部分的功能（最重要的）。它最重要的目的是为了确定用户的真正需求。

在我的经验中，这种方法非常的有效，原先对计算机没有丝毫概念的用户在你的原型面前往往口若悬河，有些观点让你都觉得非常的吃惊。在得到用户的需求之后，原型将被抛弃。因为原型开发的速度很快，设计方面是几乎没有考虑的，如果保留原型的话，在随后的开发中会为此付出极大的代价。

V 模型指出：

单元和集成测试应检测程序的执行是否满足软件设计的要求；

系统测试应检测系统功能、性能的质量特性是否达到系统要求的指标；

验收测试确定软件的实现是否满足用户需要或合同的要求。

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：

- （1）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；
- （2）风险分析：分析评估所选方案，考虑如何识别和消除风险；
- （3）实施工程：实施软件开发和验证；
- （4）客户评估：评价开发工作，提出修正建议，制定下一步计划。