

## 开始标准:

---

- 1) 开发编码结束，开发人员在开发环境中已经进行了单元测试，即开发人员完成自测。
- 2) 软件需求上规定的功能都已经实现。如果没有完全实现，开发人员提供测试范围。
- 3) 测试项目通过基本的冒烟测试，界面上的功能均已经实现，符合设计规定的功能。
- 4) 被测试项目的代码符合软件编码规范并已通过评审。
- 5) 开发人员提交了测试申请并提供了相应的文档资料。

## 停止标准:

---

- 1) 基本功能测试出现致命问题，导致50%的用例无法执行；
- 2) 版本质量太差，60%的用例执行失败；
- 3) 测试环境出现故障，导致测试无法执行；
- 4) 其他突发事件，如需要优先测试其他产品，测试人员有其他优先级更高的任务，可以申请暂停测试。
- 5) 测试人员进行冒烟测试时发现重大缺陷，导致测试无法正常进行，需要暂停并返回开发。
- 6) 测试人员进行冒烟测试时发现Bug过多，可以申请暂停测试，返回开发。
- 7) 测试项目需要更新调整而暂停，测试工作也要相应暂停。

## 完成标准:

---

- 1) 测试项目满足客户需求。
- 2) 所有测试用例都已经通过评审并成功执行。
- 3) 测试覆盖率已经达到要求。
- 4) 所有发现的缺陷都记录在缺陷管理系统。
- 5) 一、二级错误修复率达到100%。
- 6) 三、四级错误修复率达到了95%。
- 7) 所有遗留问题都有解决方案。
- 8) 测试项目的功能、性能、安全性等都满足要求。
- 9) 完成系统测试总结报告。

## alpha和beta的区别

---

含义：

Alpha测试是一种非正式验收测试，是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的测试。

Beta测试是一种验收测试，是软件产品完成了功能测试和系统测试之后，在产品发布之前所进行的软件测试活动。

测试环境的不同：

Alpha测试是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试。

与Alpha测试不同，开发者通常不在Beta测试的现场，因Beta测试是软件在开发者不能控制的环境中的“真实”应用。

测试阶段不同：

Alpha测试是软件测试过程中的第一阶段，一般只供内部测试使用；

Beta测试是软件测试过程中的第二个阶段，已经消除了软件中大部分的不完善之处。

## 开发模式：

---

现代：

---

封闭式：

在有限时间内与世隔绝的进行代码开发。

迭代式：

迭代式开发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。

每次只设计和实现这个产品的一部分，

逐步逐步完成的方法叫迭代开发，

每次设计和实现一个阶段叫做一个迭代。

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。

迭代式开发的优点：

- 1、降低风险
- 2、得到早期用户反馈
- 3、持续的测试和集成
- 4、使用变更
- 5、提高复用性

## 增量式：

是一种 进化式 的开发过程，也称为有计划的产品改进模型。它从 一组给定的需求 开始，通过构造一系列可执行中间版本来实施开发活动。第一个版本纳入一部分需求，下一个版本纳入更多的需求，依此类推，直到系统完成。每个中间版本都要执行必需的过程、活动和任务。它是原型模型和瀑布模型的结合。与原型模型不同之处：它强调每一个增量均发布一个可操作 产品，它不需要等到所有需求都出来，只要某个需求的增量包出来即可进行开发。增量模型的特点：

在稳定的框架和平台上，来开发和增加具体的业务功能

每个增量之间相对独立，各个增量可以并行开发

增量内部是瀑布模型

优点：

人员分配灵活，一开始不需要投入大量人力

先推出核心的产品，在后续增加相应的功能

增量能够有计划的管理技术风险

适用于需求经常变更的软件开发过程

缺点：

如果增量包之间存在相交的情况未很好的处理，则必须做全盘的系统分析。

## 敏捷式：

敏捷开发 是一种从1990年代开始逐渐引起广泛关注的新型软件开发方法，是一种能应对快速变化需求的软件开发能力。相对于"非敏捷"，它更强调程序员团队与业务专家之间的紧密协作、面对面的沟通（认为比书面的文档更有效）、频繁交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法，也更注重在软件开发中人的作用。敏捷软件开发主张：适度的计划、进化开发、提前交付 与 持续改进，并且鼓励快速与灵活的面对开发与变更。

四个原则：

**递增**，而不是连续的：如果开发实践是真正的敏捷精神，那么交付的工作软件是一小部分一小部分递增的。不必等到一个阶段完全完成后才开始另一个，工作也不是向大的发布日期而努力。完成的工作，但并不是业务最终期限，驱动着敏捷交付。但敏捷精神也承认业务操纵着最后截止日期。

**避免不必要的开销：**如果实践仍然是真正的敏捷精神，那么团队就致力于尽可能多地减少项目计划和文档。与其讨论要做什么，然后再写下来，不如赶紧动手去做，否则，就是在浪费时间在工作的工作上。在工作对工作中，敏捷精神有利于实际的工作交付工作软件。而且它也值面对面的交流通过邮件和其他书面文件。

**协作：**根据需求，团队成员一直与其它人进行交互，以及一些外部利益相关者。在敏捷教练世界中，整个团队的负责人Lisa Crispin能够解决所有问题，在问题出现之前。真正的敏捷精神团队是自助的。他们分配需要做的工作。虽然每个成员承担的任务都在他们的专业技能范围内，他们还是需要与团队协作的。没有人的工作是孤立的，也没有团队本身是独立工作的。没有业务利益相关者，以及诸如用户体验方面的外部专家的重大投入，团队就不可能使项目向前发展，

**说真话：**为了保证真正的敏捷，团队探讨的与项目相关的一切都要是真实的。在一些至关重要的专业领域，如冲刺测试的编码技能，他们承认存在差距。关于实际生产力，他们的要讲事实；这也就是说，在y时间内，团队是否有能力做到x。他们承认错误。说真话是一项挑战，因为他们害怕承认缺点会让他们显得很弱。但敏捷精神知道说出事实需要勇气。承认问题需要信心，然后快速地去解决问题。

敏捷开发的方法有时候被误认为是无计划性和纪律性的方法，实际上更确切的说法是敏捷开发强调适应性而非预见性。适应性的方法集中在快速适应现实的变化。当项目的需求起了变化，团队应该迅速适应。这个团队可能很难确切描述未来将会如何变化。

对比增量模型：

两者都强调在较短的开发周期提交软件，但敏捷开发的周期可能更短，并且更加强调队伍中的高度协作。

对比瀑布模型：

两者没有很多的共同点，相对来讲，敏捷方法强调的是能将尽早将尽量小的可用的功能交付使用，并在整个项目周期中持续改善和增强。。

## 传统开发模式：

---

**瀑布模型：**

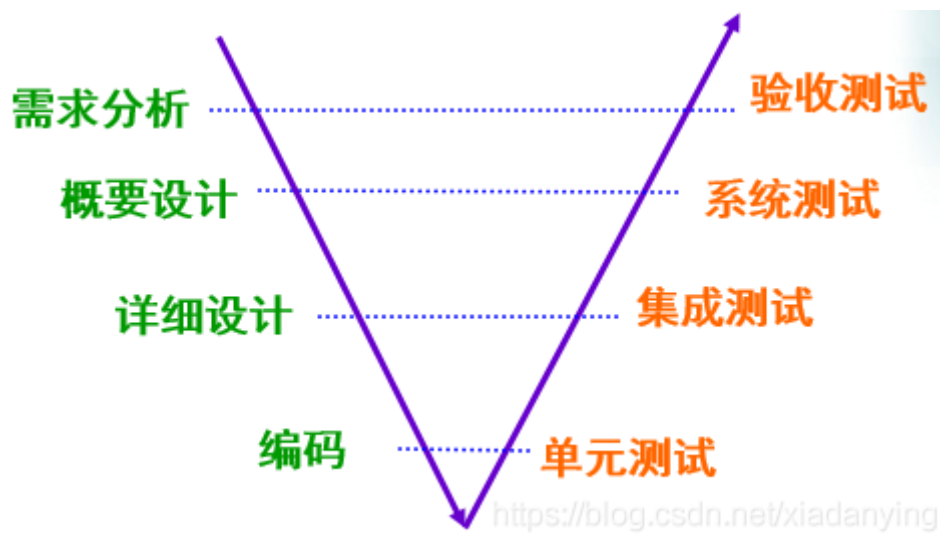
**技术特点：**

1.简单，分阶段，阶段间存在因果关系，各个阶段完成后都有评审，允许反馈，不支持用户参与，要求预先确定需求。

**使用范围：**

1.需求易于完善定义且不易变更的软件系统。

**V模型：**



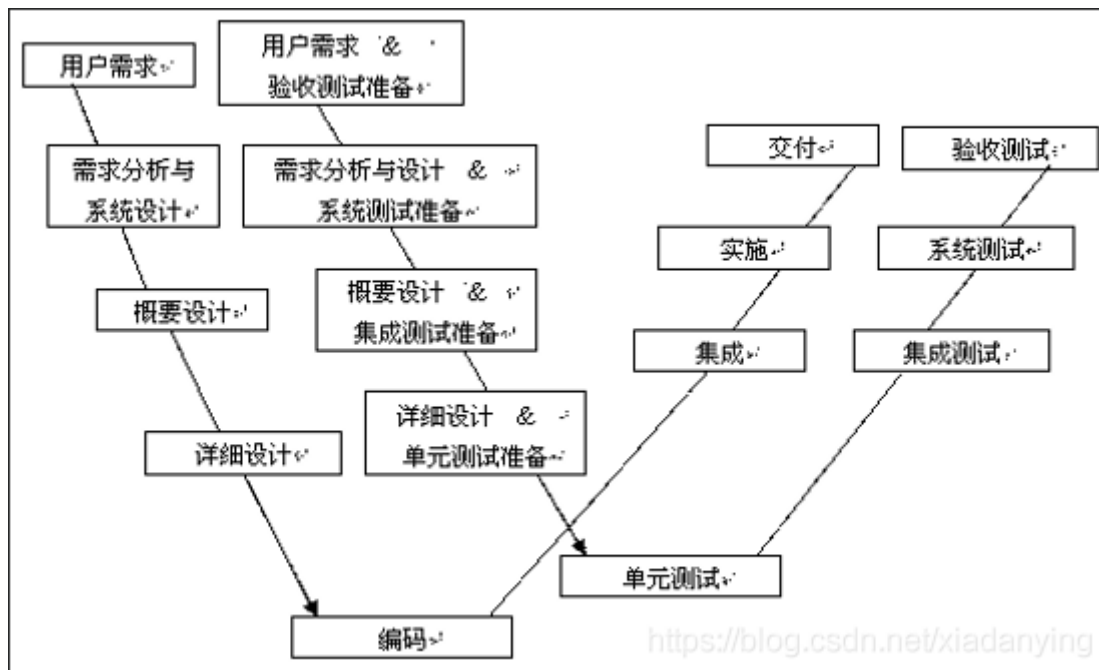
### 优点：

- 1.它非常明确地标明了测试过程中存在的不同级别，并且清楚地描述了这些测试阶段和开发过程期间各阶段的对应关系。

### 缺点：

- 1.仅仅把测试过程作为在需求分析、系统设计及编码之后的一个阶段。
- 2.测试的对象仅仅是程序，忽略了测试对需求分析，系统设计的验证，一直到最后的验收测试才被发现。

### W模型：



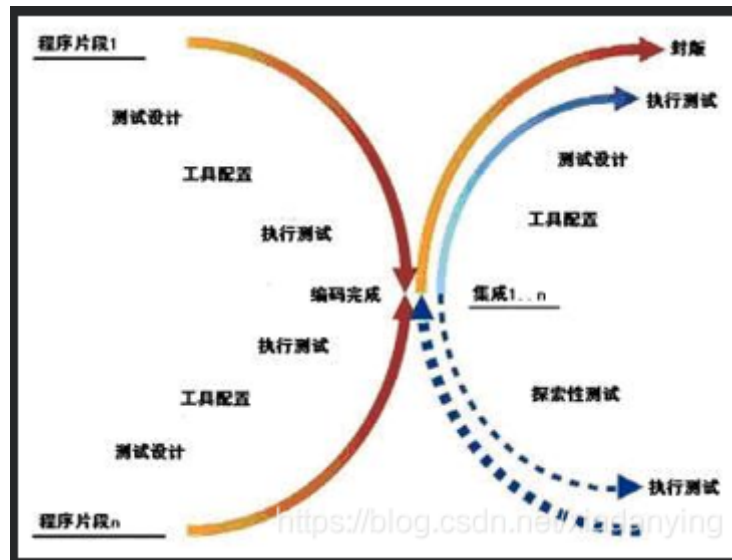
## 优点：

- 1.测试伴随着整个软件的开发周期。
- 2.测试的对象不仅仅是程序，需求、设计和功能同样要测试。

## 缺点：

- 1.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。

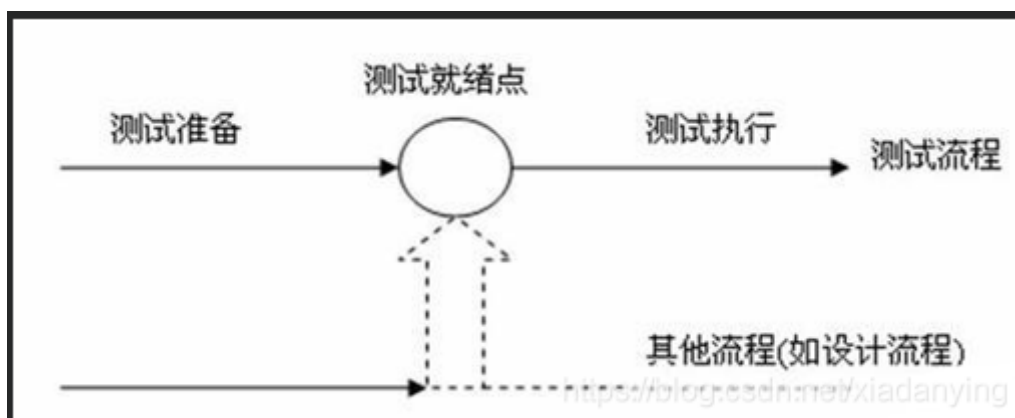
## X模型：



## 特点：

- 1.X模型的左边描述的是针对单独程序片段所进行的相互分离的编码和测试，此后将进行频繁的连接，通过集成最终合成为可执行的程序。
- 2.右上半部分，这些可执行程序还需要进行测试。已通过集成测试的成品可以进行封版并提交给用户，也可以作为更大规模和范围内集成的一部分。
- 3.X模型还定位了探索性测试（右下方）。

## H模型：



## 特点：

- 1.软件测试是一个独立的流程,贯穿产品整个生命周期,与其他流程并发地进行。H模型指出软件测试要尽早准备, 尽早执行。

## 测试：

---

### 冒烟测试：

#### 概念：

冒烟测试是在软件开发过程中的一种针对**软件版本包的快速基本功能验证策略**，是对软件基本功能进行确认验证的手段，并非对软件版本包的深入测试。冒烟测试也是针对软件版本包进行详细测试之前的预测试，执行冒烟测试的主要目的是快速验证软件基本功能是否有缺陷。**如果冒烟测试的测试例不能通过，则不必做进一步的测试。**进行冒烟测试之前需要确定冒烟测试的用例集，对用例集要求覆盖软件的基本功能。这种版本包出包之后的验证方法通常称为软件版本包的门槛用例验证。

#### 注意事项：

##### 1.注重效率：

冒烟测试应以**效率为先**，尽量缩短测试时间提高测试效率。要在关注主流程、重点功能的前提下，抓关键缺陷验数据准备，对于诸如页面不美观、用户体验不佳等缺陷可在冒烟阶段有选择的予以过滤。例如：测试系统登录，关注点应针对用户名、密码、校验码的输入及提交完成，对于非法字符的校验、登录框是否美观、错误提示是否准确等均属于次要关注点，不纳入冒烟测试范围。

##### 2.开发协同

冒烟测试阶段有几个特点，一是该阶段软件可能存在较多缺陷，特别是阻塞型缺陷，测试工作随时可能陷入停滞状态；二是该阶段测试人员对软件的流程、功能等熟悉程度较低，难免会出现找不到合适的测试方法甚至是找不到功能模块的情况从而延迟测试进度；三是该阶段的时间一般仅占整个软件生命周期的极小部分，这就需要开发人员实时响应，尽快解决各类问题。因此，在冒烟测试阶段，测试人员与开发人员的协同工作十分重要。

### 猴子测试：

#### 概念：

所谓的猴子测试（Money Test），也称搞怪测试，怪用测试，指在软件测试中，测试者可以进行各种稀奇古怪的操作模式，用以测试软件的稳定度。

猴子测试，通俗来讲是一种系统对信号因子输入稳健性的测试方法。一般用于计算机软件程序这样的逻辑严密性要求高的系统。

猴子测试就是百般刁难，乱按一通，系统也不能宕机或者数据出现差错，这样才能称得上是经得起考验的程序。

一般在产品周期的后半段，当我们在想我们已经找到了所有的可恶的bug时，猴子测试就能帮助我们提高信心。运行猴子测试几天的时间而没有引起错误能让我们从另外一个角度来判断程序的稳定性。

## 回归测试：

### 定义：

- 1.回归测试是指重复以前的全部或部分的相同测试。
- 2.新加入测试的模组，可能对其他模组产生副作用，故须进行某些程度的回归测试。
- 3.回归测试的重心，以关键性模组为核心。

通俗的说：

我发现了一个bug,扔给了开发,开发经过一段时间修复好了,通知我可以测试了,我对以前测过的有问题的那个bug再次展开测试,这个动作就叫回归测试