基本概念：

[**测试**](http://lib.csdn.net/base/softwaretest)是为了发现程序中的错误而执行程序的过程

[**软件测试**](http://lib.csdn.net/base/softwaretest)工程师在一家软件企业中担当的是“质量管理”角色，及时纠错及时更正，确保产品的正常运作

据了解，软件测试人员必须具有创新性和综合分析能力，必须具备判断准确、追求完美、执着认真、善于合作的品质，以及具有丰富的编程经验与查检故障的能力。

详细分类：

根据测试目的的不同，还有回归测试、压力测试、性能测试等，分别为了检验修改或优化过程是否引发新的问题、软件所能达到处理能力和是否达到预期的处理能力

角度细分

  从是否关心软件内部结构和具体实现的角度划分

A.白盒测试

B.黑盒测试

C.灰盒测试

  从是否执行程序的角度

A.静态测试

B.动态测试。

阶段细分

  从软件开发的过程按阶段划分有

A.单元测试

B.集成测试

C.确认测试

D.系统测试

E.验收测试

\* 单元测试：集中对用源代码实现的每一个程序单元进行测试，检查各个程序模块是否正确地实现了规定的功能。

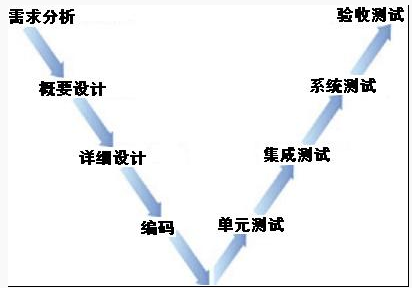
\* 集成测试：把已测试过的模块组装起来，主要对与设计相关的软件体系结构的构造进行测试。

\* 确认测试：则是要检查已实现的软件是否满足了需求规格说明中确定了的各种需求，以及软件配置是否完全、正确。

\* 系统测试：把已经经过确认的软件纳入实际运行环境中，与其它系统成份组合在一起进行测试。

测试模型(配图)：

V模型



定义：是软件开发瀑布模型的变种，它反映了测试活动与分析和设计的关系 。

描述：从左到右，描述了基本的开发过程和测试行为，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 。

左边依次下降的是开发过程各阶段，与此相对应的是右边依次上升的部分，即各测试过程的各个阶段。

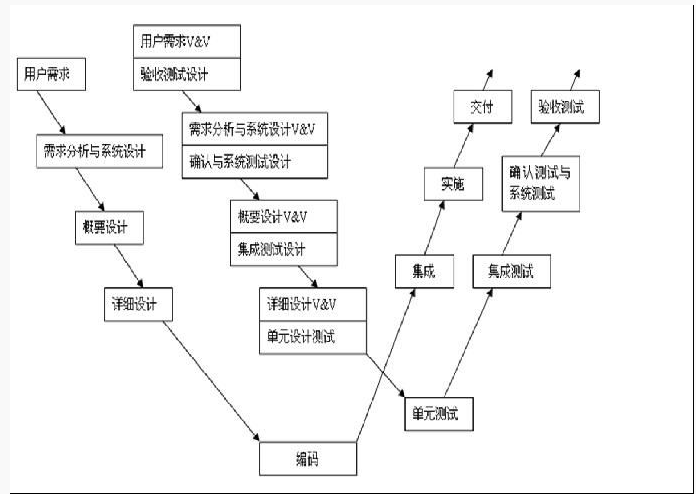
用户需求 验收测试

需求分析和系统设计 确认测试和系统测试

概要设计 集成测试

详细设计 单元测试

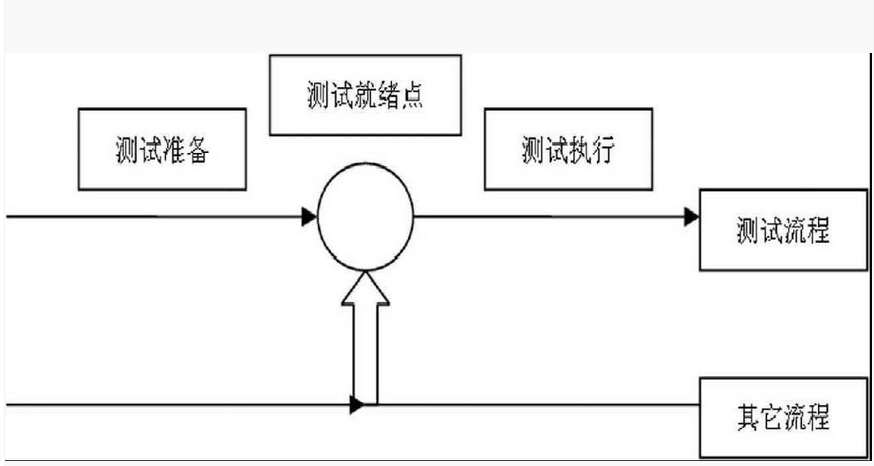
W模型



定义：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。

描述：W模型由两个V字型模型组成，分别代表测试与开发过程，图中明确表示出了测试与开发的并行关系。 W模型强调：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

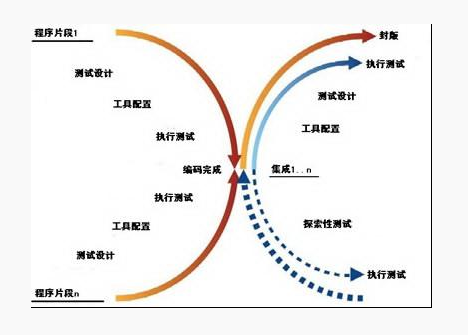
H模型



H模型中， 软件测试过程活动完全独立，贯穿于整个产品的周期，与其他流程并发地进行，某个测试点准备就绪时，就可以从测试准备阶段进行到测试执行阶段。软件测试可以尽早的进行，并且可以根据被测物的不同而分层次进行。

这个示意图演示了在整个生产周期中某个层次上的一次测试“微循环”。图中标注的其它流程可以是任意的开发流程，例如设计流程或者编码流程。也就是说， 只要测试条件成熟了，测试准备活动完成了，测试执行活动就可以进行了

H模型揭示了一个原理：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。不同的测试活动可以是按照某个次序先后进行的，但也可能是反复的，只要某个测试达到准备就绪点，测试执行活动就可以开展。



X模型

X模型也是对V模型的改进，X模型提出针对单独的程序片段进行相互分离的编码和测试，此后通过频繁的交接，通过集成最终合成为可执行的程序。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，此后将进行频繁的交接，通过集成最终成为可执行的程序，然后再对这些可执行程序进行测试。己通过集成测试的成品可以进行封装并提交给用户，也可以作为更大规模和范围内集成的一部分。多根并行的曲线表示变更可以在各个部分发生。由图中可见，X模型还定位了探索性测试，这是不进行事先计划的特殊类型的测试，这一方式往往能帮助有经验的测试人员在测试计划之外发现更多的软件错误。但这样可能对测试造成人力、物力和财力的浪费，对测试员的熟练程度要求比较高

测试工具：

总的 来说分为功能测试工具、性能测试工具、测试管理工具．

测试管理工具有td，qc，jira，bugzilla

缺陷管理工具，像TD阿，bugzilla阿，mantis

五类测试工具:负载压力测试工具、功能测试工具、白盒测试工具、测试管理工具、测试辅助工具

  负载压力测试工具

这类测试工具的主要目的是度量应用系统的可扩展性和性能，是一种预测系统行为和性能 的自动化测试工具

  功能测试工具

其主要目的是检测应用程序是否能够达到预期的功能并正常运行

  白盒测试工具

一般是针对代码进行测试，测试中发现的缺陷可以定位到代码级。根据测试工具原理的不同，又可以分为静态测试工具和动态测试工具

  测试管理工具

对测试需求、测试计划、测试用例、测试实施进行管理，并且测 试管理工具还包括对缺陷的跟踪管理

  测试辅助工具

这些工具本身并不执行测试，例如它们可以生成测试数据，为测试提供数据准

测试用例

目的：统一测试用例编写的规范，以保证使用最有效的测试用例，保证测试质量。

范围：适用于公司对产品的业务流程、功能测试测试用例的编写

术语解释

测试分析：对重要业务、重要流程进行测试前的分析。

业务流程测试用例：关于产品业务、重要流程的测试用例

测试用例设计的方法:等价类划分法、边界值分析法

测试用例设计的原则：全面性、正确性、仿真性、可操作性

测试方法：

1、等价类法

定义：是把所有可能的输入数据，即程序的输入域划分成若干部分(子集)，然后从每一个子集中选取少数具有代表性的数据作为测试用例。该方法是一种重要的，常用的黑盒测试用例设计方法

划分等价类：有效等价类、无效等价类

  1)有效等价类

是指对于程序的规格说明来说是合理的、有意义的输入数据构成的集合。利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

  2)无效等价类

与有效等价类的定义恰巧相反。无效等价类指对程序的规格说明是不合理的或无意义的输入数据所构成的集合。对于具体的问题，无效等价类至少应有一个，也可能有多个。

设计测试用例时，要同时考虑这两种等价类。因为软件不仅要能接收合理的数据，也要能经受意外的考验，这样的测试才能确保软件具有更高的可靠性。

静态测试：

（1）代码检查：代码会审、代码走查、桌面检查

（2）静态结构分析

（3）代码质量度量

动态测试：

（1）黑盒测试：又称功能测试。这种方法把被测软件看成黑盒，在不考虑软件内部结构和特性的情况下测试软件的外部特性。

     黑盒测试技术主要有等价类划分法、边界值法、因果图法、状态图法、测试大纲法以及各类典型的软件故障模型等；

（2）白盒测试：又称结构测试。这种方法把被测软件看成白盒，根据程序的内部结构和逻辑设计来设计测试实例，对程序的路径和过程进行测试

     白盒测试的主要技术有语句覆盖、分支覆盖、判定覆盖、基本路径覆盖等

心理依据：

程序测试的过程具有破坏性：

不要只是为了证明程序能够正确运行而去测试程序。相反，应该一开始就假设程序中隐藏着错误（这种假设几乎对所有的程序都成立），然后测试程序，发现尽可能多的错误

事实上，如果把测试目标定位于要证明程序中没有缺陷，那么就会在潜意识中倾向于实现这个目标。也就是说，测试人员会倾向于挑选那些使程序失效的可能性较小的测试数据。另一方面，如果把测试目标定位于要证明程序中存在缺陷，那么就会选择一些容易发现程序缺陷的测试数据。而后一种态度会比前者给程序增加更多的价值。

具体内容：

软件测试主要工作内容是验证和确认

验证(verification)是保证软件正确地实现了一些特定功能的一系列活动， 即保证软件以正确的方式来做了这个事件(Do it right)

确认(validation)是一系列的活动和过程，目的是想证实在一个给定的外部环境中软件的逻辑正确性。即保证软件做了你所期望的事情。(Do the right thing)

软件测试的对象不仅仅是程序测试，软件测试应该包括整个软件开发期间各个阶段所产生的文档，如需求规格说明、概要设计文档、详细设计文档，当然软件测试的主要对象还是源程序