

1、什么是软件测试

- 软件测试是操作代码，发现软件中隐藏的缺陷和问题，以达到保证软件质量目的工作
- 对于软件测试的几种不同看法：
 - 软件测试是检查软件产品是否符合设计要求
 - 测试是验证软件产品需求，设计和实现的一致性
 - 测试是确认产品是否满足用户的需求
 - 是对软件的全面评估
 - 投入较少的成本，达到软件产品质量最优

2、软件的生命周期

- **软件计划与可行性研究阶段：**

此阶段是软件开发方与需求方共同讨论，确定软件的开发目标和行性

- **需求分析阶段：**

确定软件开发可行的情况下，对软件需要实现的各个功能进行详细分析。

在整个软件开发过程中时不断变换和深入的阶段。

产出：《需求规格说明书》SRS

- **软件设计阶段：**

主要是根据需求文档的结果，对整个软件进行整体的设计，如：数据库，服务器，框架等

产出：

《概要设计说明书》

《详细设计说明书》

- **软件编码阶段**

将软件设计的结果转换成计算机可以运行的程序代码。在程序编码中需要保持代码统一定制。

《概要设计说明书》HLD：各个模块之间的数据交互，输入和输出

《详细设计说明书》LLD：各个模块的内部功能细节，精确到代码和代码结构

- **软件测试阶段**

在软件编码完成有，需要经过严密的测试，以发现软件在整个设计过程汇总存在的问题并加以纠正

- **软件的运行和维护阶段**

这个阶段是生命周期中最长的一个阶段，包括纠错性维和和改进性维护两个方面

3、测试的过程分类

单元测试--UT

- 单元：软件最小的一个模块，或一个代码块。
- 单元测试是开发人员自己做的测试。
- 单元测试使用的是 白盒测试方法
- 测试依据：《详细设计说明书》

集成测试--IT

- 集成测试是把一个个单元集成一个组件或者一个模块又或者一个系统
- 集成测试的测试人员：开发或者测试人员
- 集成测试的内容：虽然单元测试可以保证单元本身没有问题，但是并不代表两个单元集成到一起不会出现问题。集成测试是对两个单元的交互性测试。
- 集成测试的方法：黑盒测试
- 集成测试的依据：《概要设计说明书》

系统测试--ST

- 经过集成后的系统，对整个系统层面上的测试就是系统测试。包括功能测试，性能测试等
- 系统测试的依据：《需求规格说明书》

系统整合测试--SIT

含有硬件和第三方软件系统的测试，更接近真实用户的使用环境

验收测试---UAT

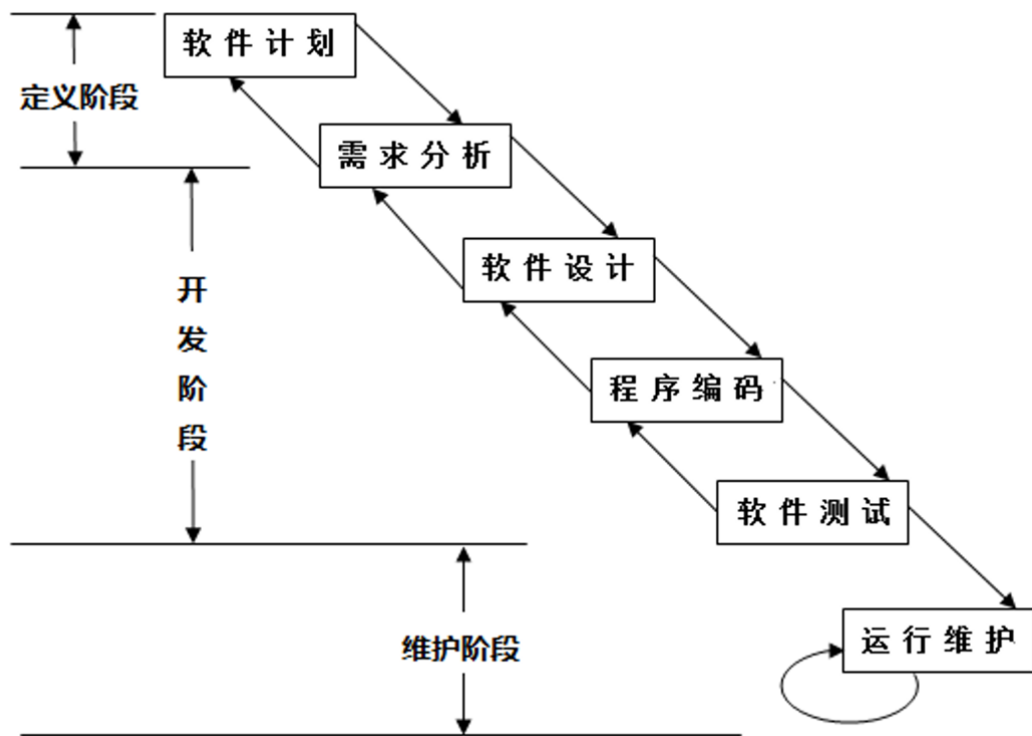
- 验收测试是指：经过甲方、产品或用户的验收的测试。验收测试是最后一步进行的测试，主要是验证软件系统是否按照合同或需求去实现功能
- 验收测试的人员：甲方、产品或用户
- 正式验收：正式验收，交付性测试
- 非正式验收：甲方或验收人员到开发环境中进行基本功能的验收

4、狭义测试和广义测试

狭义测试

动态运行程序而进行测试，测试只是在编程之后的阶段，这也是由传统的瀑布模型来决定的

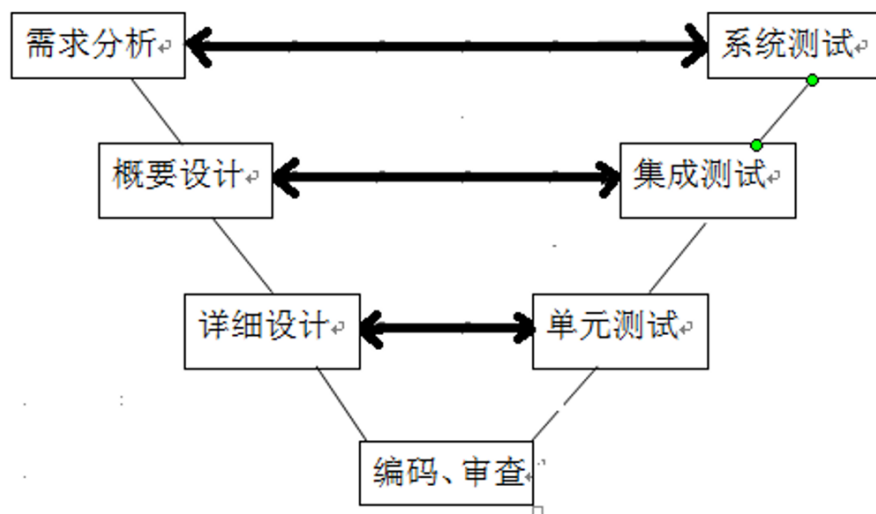
- 瀑布模型
 - 缺点：
 - 测试滞后，到最后面才介入测试，很多早期的bug没有发现
 - 对需求的适应性差
 - 优点：
 - 流程清晰明了



• V模型

基于瀑布模型的演变，认为测试是开发后的串行活动

- 优点：测试和开发对应，串行
- 缺点：文档量大，忽略了测试对象是软件本身



广义测试

静态测试+动态测试的结合，将需求评审、设计评审、代码评审等也纳入到软件测试的范畴之中。这也使得软件测试不再停留在编码之后，而成为测试贯穿整个软件周期的活动

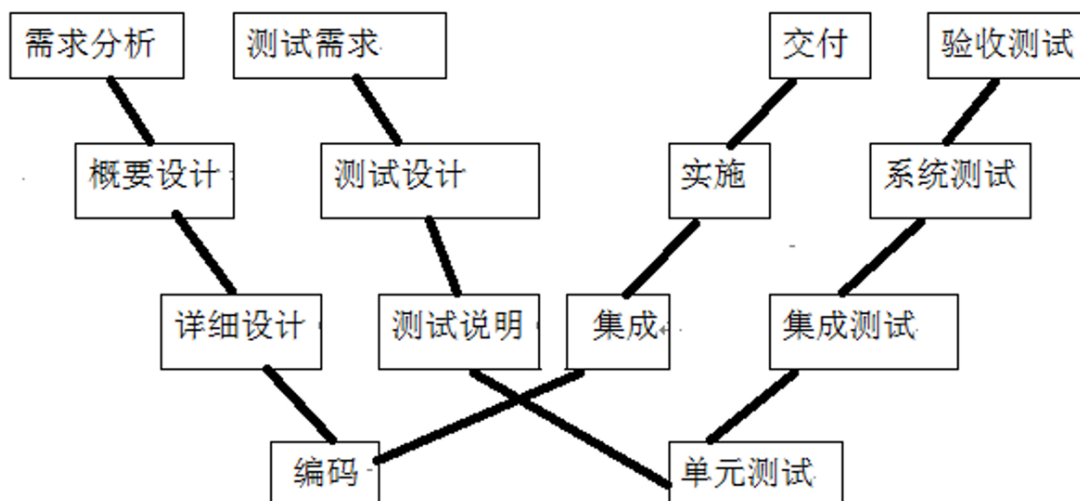
• W模型

基于V模型进行了演变

引入了静态测试，在静态测试中可以尽早的发现bug，把问题消灭在萌芽之中，会将每个阶段产生的问题及时清除，极大的提高了软件的质量，降级企业成本。

优点：测试贯穿了整个生命周期，与开发并行

缺点：对于有些项目开发过程中某个阶段没有相应的文档产生，导致W模型无法使用



5、软件的测试流程

• H模型

主要分为两个阶段：

- 测试准备阶段：
- 测试执行阶段：



• 敏捷开发模型

敏捷开发模型会在每次迭代中新增一小部分功能，迭代的速度很快，可以随时推倒重构等。

6、测试用例

测试人员根据需求预先编写包含有标题，预置条件，操作步骤和预期结果的一组案例，在执行测试用例的时候，根据用例去进行测试。

• 测试编号：

用例的编号，是唯一的，相当于给每一条测试用例加上一个独一无二的编号

Notepad_ST_Openfile_01

测试用例的编号可以根据实际情况来编写，如果项目有规定则用项目的，如果没有规定则按照自己的想法编一个

- **测试项目：**

被测用例的所属模块，对测试用例进行分类

如：B站登录，淘宝购物车，QQ登录

- **测试标题：**

这条测试用例的测试重点，或测试的方法

使用一句简单的话表述整条测试用例的内容

- **优先级：**

在测试时，决定该测试用例是否优先执行

测试用例高，需要优先执行

等级一般分为三个：高，中，低（p0,p1,p2）

高：一般是核心功能或核心流程的测试用例

中：一般的功能或普通功能的反向测试用例

低：一些页面上的优化功能

- **测试数据：**

测试步骤执行的时候需要用到测试数据，不同的测试数据会造成不同的结果

- **操作步骤：**

需要达到的预期结果要执行所有步骤

表述要清晰，明确，书面化用户

重点注意：操作步骤是动作，不能写结果

- **预期结果：**

预期达到的结果

要与实际结果进行比对，如果实际结果与预期不符，则是一个bug。

7、测试执行

冒烟测试

- 冒烟测试是验证系统主要功能和主要流程是否已经实现。如：安装，卸载，覆盖安装等
- 什么时候开始冒烟测试：开发写好代码后，提交给测试（提测/转测）的时候，测试进行冒烟测试。
- 冒烟测试通过：项目会有一个标准，例如，冒烟测试用例通过率为90%

回归测试

- 对已经测试过的功能再次进行测试就叫回归测试。
- 开发改了代码，开发修复了bug，需要回去测试上一个版本
- 回归测试的策略：
 - 再次测试全部原有功能 ----很少
 - 基于风险，测试风险较大或主要的功能

- 基于剖面，开发改了那些功能的代码，按照改动的模块再测试

探索性测试

- 不按照测试用例执行，在测试过程中根据自己的经验快速的验证产品功能

随机测试

- 完全随机测试，按照个人的想法个人对软件的理解和经验的测试，这种方法通常会发现很多意想不到的bug

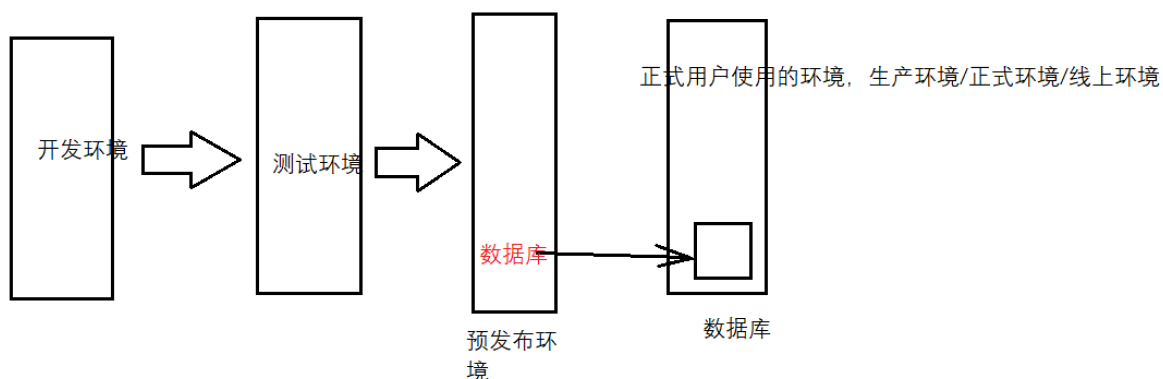
8、上线

验收测试

- 在项目上线之前会进行验收测试
- α测试：测试过程中有用户参与测试，主要测试原有功能和需求是否完全实现
- β测试：上线已有用户进行测试，好处就是有更多的测试场景。内测

预发布环境

这个预发布环境中，一般会连接生产环境的数据库，使用生产环境的数据来进行测试。



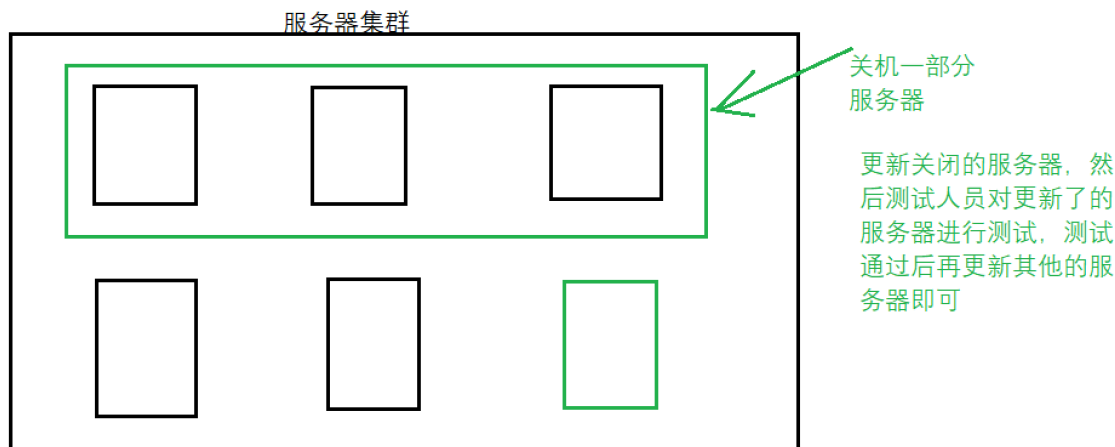
灰度发布

预发布环境测试通过后，可以进行灰度发布。

一般情况下一个项目中会有多台服务器，

先关闭其中的一部分服务器更新并进行测试，测试通过后再更新其他的服务器代码

跳板机/堡垒机：在一台服务器中间架设了一台服务器，技术人员通过中间的服务器去操作另外一台服务器



在线测试（生产环境）

代码部署到生产环境之后，测试人员需要在生产环境中进行测试

测试的内容：项目的核心流程，和新增的功能流程

生产上的数据怎么办：开发会吧生产的测试数据删除

A/B测试

A/B测试是指两种测试方案，一个是老版本，一个是改动比较大的新版本

APP测试偏多：部分app用户可以直接更新最新版本，在线上环境收集用户的反馈，主要是收集用户对新版本的使用情况，是否能够接收新版本，对新版本是否有很对的意见。如果没有太多的意见，那么会给其余没有更新最新版本的用户推送升级提示。如果新版本用户的意见很大，将新版用户回退到原有的老版本。

后续版本的测试

- EP测试：
 - 上线后，有一些比较严重的bug，用户体验不好的bug。专门针对修复这些bug的版本的测试叫做EP测试
 - 这中版本的测试没有需求文档，没有需求的审查。EP测试的目的比较明显，就是将线上的bug修复完成，并且不会应为修复了bug而引入了新的bug。
 - 测试需要和开发沟通，改动的bug会影响到那些模块。
- SP测试：
 - 小版本的测试，可能是修复一小部分的bug，也夹杂着一些小需求
 - SP测试相对要复杂一下，一方面需要验证bug，另一方面需要测试新的功能需求。但是这种版本工作量不大，时间会比较紧迫。保证没有对原有功能引入新的bug。
- 更高版本测试：
 - 一般是指大版本测试，如10.5更新到11.0。接班就是一个新版本的测试流程，但是修改比较大，需求多，回归测试工作量大，1-3个月做一个版本。

9、全程测试

- 测试的左移和右移
 - 测试左移：测试向开发移动。测试和开发进行单元测试，代码走读，代码审计。如：开发在写代码是，测试利用测试思维告诉开发，将未来可能出现的bug消灭在编码阶段
 - 测试右移：向运维移动。利用一些自动化机技术，部署到环境中监控线上环境的稳定性等。