了解软件测试的含义

软件测试遵循的准则

软件测试有哪些分类？分别是什么概念？

何时开始测试？测试方案如何设计？

测试流程是怎么样的？怎么提bug？怎么写报告？

为什么要做自动化？怎么做？

IEEE定义：

使用人工或自动的手段来运行或测量软件系统的过程，以检验系统是否满足规定的要求，并找出与预期的结果之间的差异

软件测试的测试对象？

需求-概要设计-详细设计-源码-可运行的程序-软件运行环境

软件测试五大要素和两个目标

质量 人员 资源 流程 技术

**一、软件测试的分类**

**1、测试阶段**

单元测试 集成测试 系统测试 验收测试

**1.1 单元测试**

对软件中的最小可测试单元进行检查和验证

**1.1.1 单元测试的原则**

① 尽可能保证各个测试用例的是相互独立的

② 一般由代码的开发人员实施，用来检验所开发的代码功能符合自己的设计要求

**1.1.2 单元测试的益处**

①能尽早发现缺陷 ②有利于重构 ③简于集成 ④文档 ⑤用于设计

**1.1.3 单元测试的限制**

① 不能覆盖所有的执行路径，所以不能保证捕捉到所有路径的错误

② 每一行代码一般需要3-5行测试代码才能完成单元测试。存在投入和产出的一个平衡

**1.1.4 单元测试框架**

① Junit 针对java ②PHPUnit 针对php

**1.2 集成测试**

在单元测试的基础上，测试将所有软件单元按照概要设计规格说明要求组装成模块、子系统或系统的过程中的各个部分工作是否达到或实现相应的技术指标及要求的活动

**1.2.1 集成测试的主要实施方案**

a. big bang （一次性集成）

b. 自顶向下 （递增 主程序向下集成测试）

c. 自顶向上 （程序模块的最底层开始测试）

d. 核心系统集成 （先对核心进行测试）

e. 高频集成

集成测试&单元测试

测试对象不同

测试的依据不同

测试方法不同 集成：接口 单元：单元内部

**1.3 系统测试**

系统业务角度

**1.4 验收测试**

针对用户需求、业务流程

用户验收测试（开发方）

运行验收测试（运维）

①alpha测试：开发环境 ②beta测试：完全脱离开发环境

**2、软件测试手段**

黑盒 白盒 静态 动态 手工 自动

**2.1 黑盒测试：**

外部结构、针对软件界面或可见的功能

用户的视角：通过不同的数据和不同的结果来判断

**2.1.1优点：**

① 容易实施，不需要关注内部的实现

② 更贴近用户的使用视角

**2.1.2缺点:**

① 测试覆盖率低，覆盖代码量的不到40%

② 针对黑盒自动化测试，复用率比较低，维护成本较高

**2.1.3 主要测试**

①是否有不正确或者遗漏的功能

②接口上输入是否正确接收，能否输出正确的结果

③数据结构是否有错误或者外部信息访问错误

④性能是否能够满足要求

**2.1.4 黑盒的主要设计方法**

等价类划分 边界值分析 错误推测 因果图 正交试验 状态迁移 流程分析

**2.2 白盒测试**

针对程序的逻辑结构进行测试

主要逻辑单位：语句、条件、条件组合、分支、路径

**2.2.1 优点**

①理解系统原理

②检测代码中每条分支和路径

③揭示隐藏在代码中的错误

④对代码的测试比较彻底

**2.2.2 缺点**

①成本昂贵，较高的覆盖率，工作量大

②无法检测代码中遗漏的路径和数据敏感性错误

③不能直接验证需求的正确性

2.2.3 主要测试方法

①代码检测法 ②静态结构分析法 ③静态质量度量发法

④逻辑覆盖法 ⑤基本路径测试

**\*\* 灰盒测试**

介于白、黑之间，关注输出对于输入的正确性，同时也关注内部表现

**2.3 静态测试**

无须执行被测程序，通过评审软件文档或代码等检查软件是否符合变成标准，借以程序的不足，减少错误

互审 走查 会议

**2.4 动态测试**

运行被测程序，检查运行结果与预期结果的差异，并分析运行效率、正确性和健壮性等

**2.5 手工测试**

专门测试人员从用户视角来验证软件是否满足设计要求

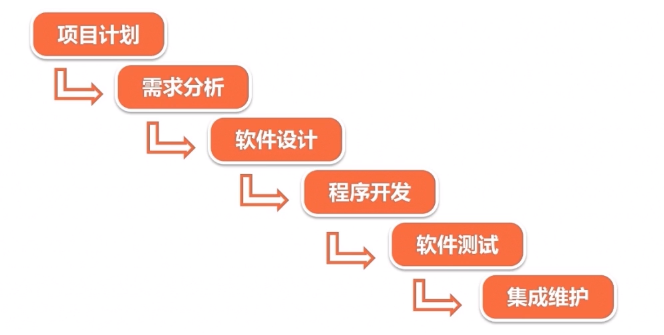
深度测试和主观性判断测试

**2.6 自动化测试**

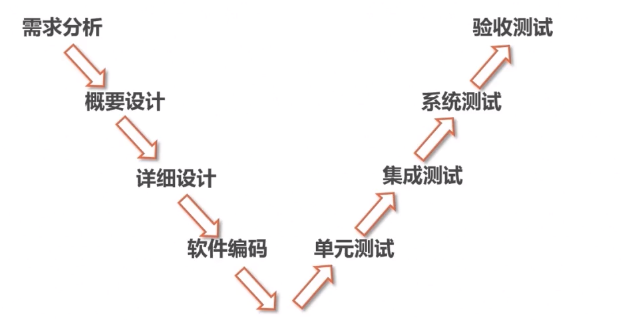
使用单独的测试工具软件控制测试的自动化执行以及预期和结果进行自动检查

**3、测试模式**

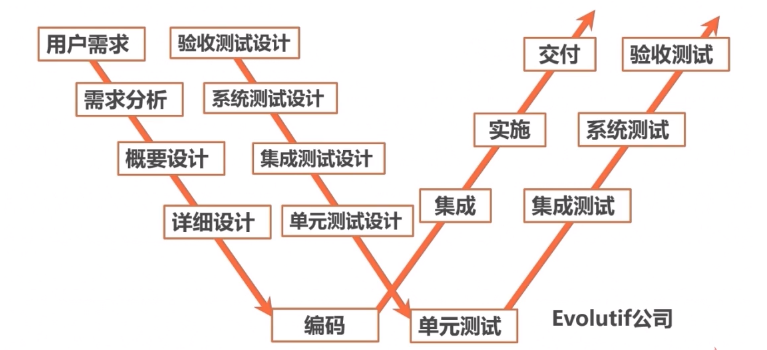
**3.1 传统瀑布模型**



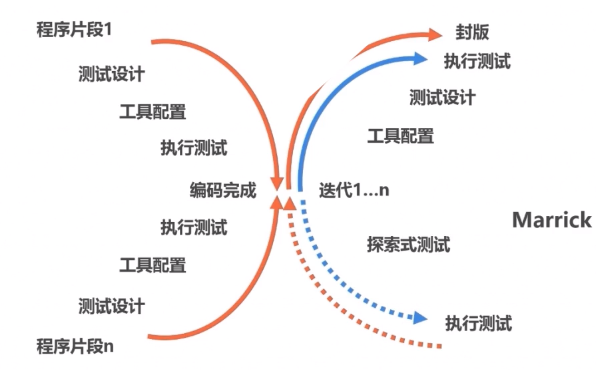
**3.2 v模型**



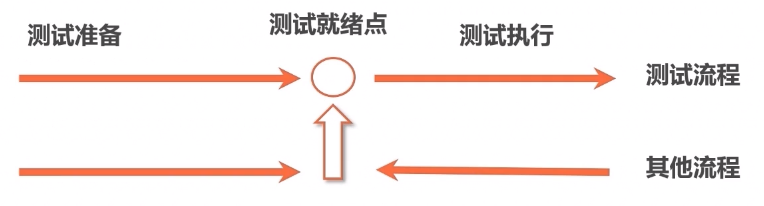
**3.3 w模型**



**3.4 x模型**



**3.5 H模型**



**3.6 敏捷测试模式**

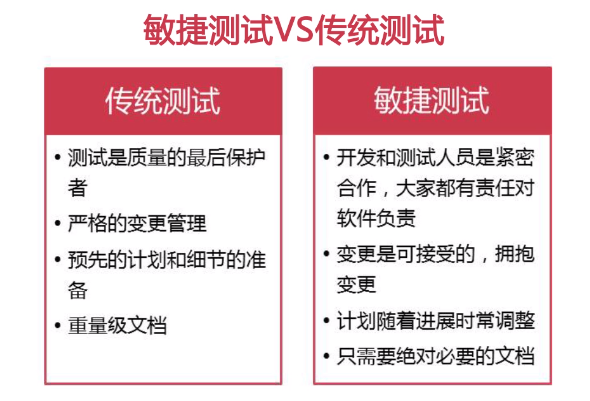
①强调从客户角度进行测试

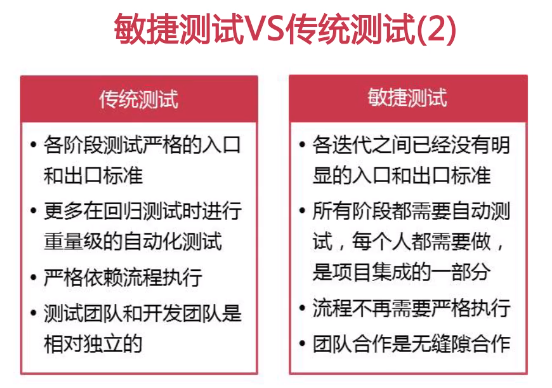
②重点关注迭代测试新功能

③尽早测试，不间断测试，具备条件即测试

④强调持续反馈

⑤预防缺陷重于发现缺陷





**3.7 基于脚本测试**

ST：选择测试设计在执行测试

ET：探索是测试 完全抛开测试脚本的测试



**3.8 基于风险测试-RBT**

Risk-based Testing

基于对软件失效的风险评估并以此指导测试计划、设计、执行、结果评价的软件测试类型

质量风险 ：软件功能 性能 功能缺失 数据

管理风险 ：关联第三方系统等

风险级别 = 风险可能性 \* 风险严重程度



风险要素分 = Sum（单项权重\*得分）

**3.9 探索式测试**

优点：

更能激发测试人员的创造性和工作乐趣

增加新的或较深bug的可能性

在较短时间在找到更多的bug以及对sut作一个快速的评估

有利于更加有效的实施自动化

更加适用于敏捷项目

减少了在简单繁复用例的无谓编写时间

缺点：

测试管理上有局限性

对bug的重复利用和重现上作用有限

对测试人员的测试技术和业务知识深度以来较大

只有在sut已完全可用的前提下才更有作用

ET的生产率很难定义

ET本身难以进行自动化

**3.9.1全局/局部**

**局部探索式测试**

输入、状态、代码路径、用户数据、执行环境

**全局探索式测试**

漫游测试法：游客游览

**3.10 基于模型的测试-MBT**

对需求功能点建模

偏向自动化测试

主要MBT工具

spec Explorer

GraphWalker

Tcases

modeljunit

**4、软件测试类型**

功能测试

性能测试 部署测试 文档测试 安全测试

兼容性测试 易用性测试 本地化测试 无障碍测试

可靠性测试

**4.1功能测试**

**4.1.1 针对的问题**

功能错误或者遗漏

界面问题

性能错误

数据以及访问错误

初始化及终止错误

**4.1.2 功能测试工具**

QTP winrunner 功能自动化测试工具

selenium

Watir

Sikuli

**4.2性能测试**

负载测试 压力测试 稳定性测试

**4.2.1 性能指标**

web应用： 并发用户数(UV)每秒事务数(TPS) 系统相应时间 设备性能

**4.2.2 性能测试工具**

LoadRunner /Jemeter/ Apache Bench/ losdUI /

**4.2.3 静态性能评估**

页面代码 对web应用的页面进行静态分析

YSlow /PageSpeed 浏览器插件

**4.2.4 应用性能管理（APM）**

对系统实施监控，实现性能管理、故障管理的解决方案

www.tingyun.com

**4.3安全测试**

对软件产品进行测试以确保其符合产品安全需求和质量测试

渗透测试：

模拟对软件系统的恶意攻击行为来评估系统安全性的一种测试

OWASP/

Appscan / webinspect/ nessus/ nmap/MetaSploit / webScarab/ Fortify

W3AF/

**4.4兼容性测试**

软件本身的兼容性 ： 版本

不同平台下的兼容性

软件对运行设备的兼容性

软件互操作

web应用：浏览器内核（浏览器的兼容）

浏览器工具：BrowserShots / BrowserSand box

Google 浏览器兼容测试插件

**4.5文档测试**

针对软件产品的交付品、配套的文档类部件的测试，用户手册、使用说明等。

**4.5.1 文档测试关注要点**

完整性、正确性、一致性、易理解性、易浏览性

**4.6 可靠性测试**

软件可靠性 / 硬件可靠性

**4.7易用性测试**

用户使用软件时是否感觉方便，是否能保证用户使用体验的测试类型

**4.8 本地化测试**

针对软件的本地化版本实施的针对性测试

**4.8.1主要测试内容：**

语言、书写习惯

时区、日期格式、货币

当地风俗、法律法规

政治敏感内容

**4.9 部署测试**

主要验证系统部署过程中，并确保软件经过安装测试后可以正常使用

**4.9.1主要测试内容：**

在不同环境下的部署验证

参照部署文档执行，过程的合理性、正确性

基础数据

**4.10 无障碍测试**

可访问性测试/ 便于特殊人群使用的功能

**5、其他软件测试类型**

回归测试/ Monkey测试/ 冒烟测试

**A/B测试**

提供2个版本给两个用户使用并记录相关的用户行为数据，来确定更优化设计的一种测试方案

**A/B测试实施要点：**

多个方案并行、 每次测试仅改动一个变量、 按照某种规则进行优胜略汰

**A/B测试工具：**

google Analytics Content Experiments /

