

@测试定义：为了发现程序中的错误而执行程序的过程

她是帮助识别开发完成（中计算机软件（整体或部分）的正确态度、完全度和质量）的软件过程；

是 SQA（software quality assurance）的重要子域。

@测试的目标：1、测试是为了发现程序中的错误而执行程序的过程

2、好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案

3、成功的测试是发现了至今为止尚未发现的错误的测试

@测试的内容：主要是验证和确认

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)

1.确定软件生存周期中的一个给定阶段的产品是否达到前阶段确立的需求的过程。

2.程序正确性的形式证明，即采用形式理论证明程序符合设计规约规定的过程。

3.评审、审查、测试、检查、审计等各类活动，或对某些项处理、服务或文件等是否和规定的需求相一致进行判断和提出报告。

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)

@测试的分类：黑盒测试、灰盒测试、白盒测试、自动化测试、性能测试

@软件测试的意义

保证软件质量

@测试原则：1、测试应该尽早进行，最好在需求阶段就开始介入，因为最严重的错误不外乎是系统不能满足用户的需求。

2、程序员应该避免检查自己的程序，软件测试应该由第三方来负责。

3、设计测试用例时应考虑到合法的输入和不合法的输入以及各种边界条件，特殊情况下要制造极端状态和意外状态，如网络异常中断、电源断电等。

4、应该充分注意测试中的群集现象。

5、对错误结果要进行一个确认过程。一般由 A 测试出来的错误，一定要由 B 来确认。严重的错误可以召开评审会议进行讨论和分析，对测试结果要进行严格地确认，是否真的存在

这个问题以及严重程度等。

6、制定严格的测试计划。一定要制定测试计划，并且要有指导

性。测试时间安排尽量宽松，不要希望在极短的时间内完成一个高水平的测试。

7、妥善保存测试计划、测试用例、出错统计和最终分析报告，为维护提供方便。

@测试流程：需求--》测试计划--》测试需求分析--》测试用例--》测试方案--》测试执行--》回归测试--》测试报告

@测试步骤、测试阶段划分：（1）单元测试（也称模块测试）：

1、是开发编写小段代码，用于检测被测代码的一个很小的、很明确的功能是否正常。通常单元测试有程序员自己完成

2、关注点：一个模块的功能实现正确及常规错误测试

3、方法：进行白盒测试

（2）集成测试（也称组装测试，联合测试）：

1、在单元测试的基础上，将所有模块按设计要求集成在一起进行测试，以检验总体设计中各模块间的接口设计问题、模块之间的相

互影响、上层模块存在的各种差错及全局数据结构对系统的影响等方面。

2、测试人员根据编写的功能模块测试用例、子系统业务流或系统业务流，执行功能测试。

（3）系统测试：

1、经过集成测试的软件，针对整个产品的全面测试，既包含各模块功能正确性测试和各业务场景测试，包括对整个产品的健壮性（容错性）、安全性、可维护性及

各种性能参数的测试

2、针对整个产品的全面测试

3、执行健壮性测试，安全性测试，可维护性测试以及各种性能参数的测试

（4）确认测试（也称验收测试，有效性测试）：

1、是部署软件之前的最后一个测试操作。在软件产品完

成了单元测试、集成测试和系统测试后，产品发布之前锁机进行的软件测试活动。是技术测试的最后一个阶

段，也成交付测试。

2、目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务

3、关注点，软件功能是否满足用户需求，符合产品需求规格说明书

4、方法：用户根据验收计划和项目验收标准，对软件进行测试

@单元、集成、系统测试的比较：

(1) 测试方法不同：单元测试属于白盒测试

集成测试属于灰盒测试

系统测试属于黑盒测试

(2) 考察范围不同：单元测试主要测试单元内部的数据结构、逻辑控制、异常处理

集成测试主要测试模块之间的接口和接口数据传递关

系，以及模块组合后的整体功能

系统测试主要测试整个系统相对于需求的符合度

(3) 评估基准不同：单元测试的评估基准主要是逻辑覆盖率

集成测试的评估基准主要是接口覆盖率

系统测试的评估基准主要是测试用例对需求规格的覆盖率

@测试需求理解：(1) 需求理解：1、软件需求是编写测试用例的依据 2、是衡量测试覆盖率的重要指标

(2) 业务流程：1、系统业务流程 2、模块业务流程

(3) 需求检查：1、正确性 2、可测性

(4) 提取业务功能点：1、功能图

@什么是黑盒测试：1、黑盒测试把被测对象看成一个黑盒，只考虑其整体特性，不考虑其内部具体实现。

2、黑盒测试针对的被测对象可以是一个系统、一个子系统、一个模块、一个子模块、一个函数等。

3、黑盒测试又可以被称为给予规格的测试。

@常见的黑盒测试类型：1、功能性测试，是一种顺序测试每个程序特性或功能，另一种途径是一个模块一个模块的测试，即每个功能在其最先调用的地方被测试；

2、容量测试，检测软件在处理海量数据使得局限性，能发现系统效率方面的问题；

3、负载测试，检测系统在一个很短时间内处理一个巨大的数据量或执行绪多功能调用上的能力；

4、恢复性测试，主要保证系统在崩溃后能够恢复外部数据的能力；

@常用的黑盒测试方法：等价类划分法（有效等价、无效等价）、边界值分析法、判定表发、状态迁移法、因果图。。

@黑盒测试的特点：1、对于更大的代码单元来说（子系统甚至系统级）比白盒测试效率要高；

2、测试人员不需要了解实现的细节，包括特定的编程语言；

3、从用户的视角进行测试，很容易被大家理解和接受；

4、有助于暴露任何规格不一致或有歧义的问题；

@测试用例编写规范：

1、流程性

2、连贯性

3、全面性

4、正确性

@测试用例包括哪些要素：用例 ID、用例名称、测试目的、测试级别、参考信息、测试环境、前提条件、测试步骤、预期结果

@好的测试用例的依据：产品原型图、需求说明书、业务规则、业务逻辑、数据字典、概设、详设

@缺陷的相关属性、要素：缺陷发现人、缺陷发现时间、缺陷状态、缺陷严重程度、缺陷所属版本、缺陷修改日期、操作步骤

@QC 中软件缺陷状态列表：new 缺陷的初始状态

open 开发人员开始修改缺陷

fixde 开发人员修改缺陷完毕

closed 回归测试通过

reopen 回归测试失败

postpone 推迟修改

rejected 开发人员认为不是程序问题，拒绝缺陷

duplicate 与已经提交的 defect 重复

abandon 呗 reject 和 duplicate 的 defect，测试人员确认后的确不是问题，将 defect 置为此状态

@缺陷管理工具：禅道、QC（Mercury Quality Center）、JIRA、Bugzilla、Bugfree 、Bugzero 、Rational ClearQuest、Mantis

@缺陷的流程（生命周期）：发现、提交、验证、修复、关闭

@灰盒测试：1、根据利用的被测对象信息不同，会采用不同的方法进行测试。

2、利用被测对象的整体特性信息，采用黑盒测试。

3、利用被测对象的内部具体实现信息，采用白盒测试方法。

4、如果及利用被测对象的整体信息，有利用被测对象的内部具体实现信息，采用灰盒测试。两种信息占的比例不同，相应的灰度就不同。完全是整体特性

信息，就是黑盒测试，完全是内部具体实现信息，就是白盒测试。

5、典型的灰盒测试比如集成测试和系统测试时借助 log 信息。

@白盒测试：1、白盒测试是依据被测软件分析程序内部构造，并根据内部构造设计用例，来对内部控制流程进行测试，可完全不顾程序的整体功能实现情况。

2、白盒测试是基于程序结构的逻辑驱动测试。

3、白盒测试又可以被称为玻璃盒测试、透明盒测试、开放盒测试、结构化测试、逻辑驱动测试

@自动化测试：一般是指通过计算机模拟人的测试行为，替代人的测试活动，狭义上是指测试执行由计算机来完成。

@性能测试：1、性能测试：针对系统是否能够达到其规定的性能需求进行测试，如容量和响应时间（性能测试的目标就是设计和执行性能测试用例来验证这些测试对象是否达到了

预期的性能指标，如在给定参考符合条件下，通过模拟网上话务模型和业务模型，考察系统互损、时延概率、以及呼叫处理性能）

2、压力测试：压力测试通过时系统达到或者超过其设计的最大负荷来检验其在非正常状态下的特性，特性是系统所需要完成的正常功能。

@自动化测试工具：selenium2、monkey。。。。

@基线是什么：经过评审的作为入库的技术的标准。

@概要设计项的定义：1、针对需求规格说明书中的需求项作分解，一个需求项可以被分级为一个或多个模块，每个模块间有明确的接口，模块的功能独立，符合高内聚，低耦合的原则，表示

每个模块的项目及概要设计向概要设计项最终落实到概要说明书中

2 命名方式：产品编号-HLD-子系统名-模块名-XXX

如：CALC-HLD-ADD-DEL 表示计算器十进制假发功能模块

@软件需求跟踪规程

@软件需求跟踪输入：1、工作任务书

2、需求说明书、概要设计、详细设计、代码、单元测试、集成测试和系统测试文档

3、已经基线化的需求、设计文档、代码等的变更请求

@详细设计项的定义：1、为了实现概要设计说明书中的模块，再详细设计中，把概要设计模块细化到函数或函数组的层面，函数或函数组及详细设计项，详细设计项目最终落实到软件详细设计说明书中

2、命名方式：产品编号-LLD-模块名-XXX

如：CALC-LLD-ADD-DEL-DEC-001 表示计算器十进制加法功能模块参数判断函数的详细设计项

@需求的定义：1、解决用户问题或达到用户目标所需的条件或能力

2、未遵循合同、标准、规格或其他要求的正式文档，系统必须满足或拥有的条件或能力

3、按文档化表现（1）和（2）中的条件或能力，就是 SRS 强调做什么(what)而不是如何做（How）

@需求跟踪矩阵：需求--》设计--》编码--》测试

@需求工程和需求管理的关系：1、需求工程包含需求管理

2、需求管理侧重与需求工程中的管理活动

3、需求管理是 CMM 二级的第一个 KPA

@需求项的定义：1、根据工作任务书（客户需求）的规格，把任务书中的任务分解可以实现的符合要求的具体的需求项，需求项最终落实到需求文档中（SRS）

2、命名方式：产品编号-SRS-需求类型-特姓名-XXX

如：CALC-SRS-FUNC-ADD-001 表示计算器是禁止加法功能需求

@测试计划与测试方案的区别：

所谓测试计划是指描述了要进行的测试活动的范围、方法、资源和进度的文档。它主要包括测试项、被测特性、测试任务、谁执行任务和风险控制等。

所谓测试方案是指描述需要测试的特性、测试的方法、测试环境

的规划、测试工具的设计和选择、测试用例的设计方法、测试代码的设计方案。

计划：属于组织管理层面的文档，从组织管理的角度对测试活动进行规划；

方案：属于技术层面的文档，从技术的角度对测试活动进行规划。

@回归测试：软件在测试或其他活动中发现的缺陷经过修改后，应该进行回归测试。目的是验证缺陷得到了正确的修复，同时

对系统的变更没有影响以前的功能。回归测试可以发生在任何一个阶段，包括单元测试、集成测试和系统测试。

@回归测试策略：

1、完全重复测试：重新执行所有在前期测试阶段建立的测试用例，来确认问题修改的正确性和修改的扩散局部影响性

2、选择性重复测试：既有选择地重新执行部分在前期测试阶段建立的测试用例，来测试被修改的程序

3、覆盖修改法：针对被修改的部分，选取或重新构造测试用例验证没有错误再次发生的用例选择方法

4、周边影响法：该方法不但要包含覆盖修改法确定的用例，还需要分析修改的扩散影响，对那些受到修改间接影响的部分选择测试用例验证它没有收到

不良影响。该方法比覆盖修改法更充分一点。

5、指标达成法：这是一种类似于单元测试的方法，再重新执行前，先确定一个要达成的指标，如修改部分代码 100%的覆盖、与修改有关的接口 60%的覆盖

等，基于这种要求选择一个最小的测试用例集合。

@自动化测试的意义：

- 1、对程序新版本运行前一版本执行的测试，提高回归测试效率
- 2、可以运行更多更频繁的测试，比如冒烟测试
- 3、可以执行手工测试困难或不可能做的测试，比如大量的重复操作或者集成测试
- 4、更好地利用资源，比如测试仪器或者被测对象

5、测试具有一致性和可重复性，即自动化测试的步骤和结果是完全一样的

6、测试的复用性，即自动化测试脚本可以拆分开给其他测试脚本使用

7、可以更快的将软件推向市场，软件发布前进行高效的回归测试，较少软件发布的时间

8、增加软件信任度，通过自动化测试提高了测试的效率，可以节约时间

@自动化测试的限制：

1、不能取代手工测试，自动化测试只能提高测试效率，不能提高测试有效性，即不可能发现更多缺陷

2、手工测试比自动测试发现的缺陷更多

3、对测试设计依赖性极大，测试设计的不好会遗漏问题

4、自动化测试对软件开发具有很大的依赖性，开发上出现变更可能导致前面的自动化测试完全失效

5、工具本身并不具备想象力

6、不现实的期望，希望自动化取代手工测试

7、缺乏测试实践经验，手工测试都做不好，或者经验积累不够，就尝试自动化，很难成功

8、期望自动化测试发现大量新缺陷，自动化只能保证测试执行效率，确保已有的问题不会再发生，发现新缺陷不是其目的

9、安全性错觉，认为进行了自动化测试的软件就是安全的、质量有保证的

@回归测试自动化：

1、回归测试是一个重用以前成果的测试

2、回归测试的自动化法包括测试程序的自动运行、自动配置，测试用例的管理和自动输入，测试的自动执行，测试信息与结果的自动采集，

测试结果的自动采集，测试结果的自动比较和结论的自动输出，尤其前面提到的各类数据的共享决策

3、对系统测试功能比较简单、测试界面相对稳定并且测试用例良好组织的测试来说，采用“捕捉回放”工具比较合适，如：

QTP、Selenium2、Robot、SilkTest

4、实现测试用例的自动化并实现测试结果的自动判断，脚本化

的、包好控制结构、内部实现结果判断的测试用例是唯一的选择，
此类脚本语言有：TCL、Python、Perl 等

5、回归测试的自动化需要在做测试方案时就要考虑的，必要时
要投入资源进行的开发，能形成可供继承与推广的工具

@自动化环境搭建：

JDK1.6、eclipse 火狐浏览器、谷歌浏览器（chrome）、ie
selenium 的 jar 包、selenium ide 插件

@性能测试：

通过一定的手段，再多并发下情况下，获取被测系统的各项性能
指标，验证被测系统在高并发下的处理能力、相应能力，稳定性
等，能否满足预期。定位性能瓶颈，排查性能隐患，保障系统质
量，提升用户体验

@什么系统适合做性能测试：

用户量大，PV 比较高的系统、系统核心模块/接口、业务逻辑/算
法比较复杂、促销/活动推广计划

@性能测试发生的场景:

新系统，新项目、线上性能问题验证和调优、新技术选型、性能容量评估和规划、日常系统性能回归

@性能测试指标:

事务 (Transaction)

TPS(Transaction Per Second 每秒成功的事务数)

并发数/虚拟用户 (Vuser)

响应时间 (Response time)

PV(page view)

UV(Unique Visitor)

场景 (Scenario)

思考时间 (Think Time)

吞吐量 (Throughput)

常用监控指标: CPU、内存、磁盘 IO、网络 IO、线程状态、

JVM、中间件、数据库指标

@性能测试流程：

需求调研--》测试计划--》环境搭建--》数据准备--》测试脚本--》
执行测试--》调优回归--》测试报告

@a 测试：是由用户在开发环境下进行的测试，也可以是开发机构内部的用户在模拟实际操作环境下进行的测试

a 测试时，软件在一个自然设置转台下使用。开发者坐在用户旁，随时记下错误情况和使用中的问题。这是受控制的环境下进行的测试

目的主要是评价软件产品的 FLURPA（功能、局域化、可用性、可靠性、性能和技术支持）

@β 测试：是由软件的多个用户在一个或多个用户的实际使用环境下进行的测试

β 测试时开发者通常不在测试现场。因而，β 测试是在开发者无法控制的环境下进行的软件现场应用

@app 测试总结:

- 1、安装，卸载测试
- 2、在线升级测试
- 3、业务逻辑测试
- 4、异常测试
- 5、客户端性能测试
- 6、易用性测试
- 7、电量与电流测试
- 8、内存泄漏测试
- 9、外网与场景测试

比较 PC 端:

- 1、大的思路和 PC 端一样，不同的 app，比 PC 端多元化，复杂
(场景，3G，WIFI) 多元化， 复杂 (如 3G，WIFI)

(自动化在牛也代替不了人工) 不可能全自动化

接口、底层、中低层

UI 测试: 自动化实现比较头疼

crash: 次数 crash 基本信息 (crash 闪退)

APP 和 WEB 能否公用一个借口, 底层数据不同, 在 APP 端做数据过滤, 两个不同的接口不同的数据

App 测试中 ios 和 Android 有哪些区别呢? 下面是从其他网站看来的答案, 还有什么其他的区别呢? 希望大家多多补充。。。

1. Android 长按 home 键呼出应用列表和切换应用, 然后右滑则终止应用;
2. 多分辨率测试, Android 端 20 多种, ios 较少;
3. 手机操作系统, Android 较多, ios 较少且不能降级, 只能单向升级; 新的 ios 系统中的资源库不能完全兼容低版本中的 ios 系统中的应用, 低版本 ios 系统中的应用调用了新的资源库, 会直接导致闪退 (Crash);
4. 操作习惯: Android, Back 键是否被重写, 测试点击 Back 键后的反馈是否正确; 应用数据从内存移动到 SD 卡后能否正常运行等;
5. push 测试: Android: 点击 home 键, 程序后台运行时, 此时接收到 push, 点击后唤醒应用, 此时是否可以正确跳转; ios, 点击

home 键关闭程序和屏幕锁屏的情况（红点的显示）；

6. 安装卸载测试：Android 的下载和安装的平台和工具和渠道比较多，ios 主要有 app store，iTunes 和 testflight 下载；

7. 升级测试：可以被升级的必要条件：新旧版本具有相同的签名；新旧版本具有相同的包名；有一个标示符区分新旧版本（如版本号），对于 Android 若有内置的应用需检查升级之后内置文件是否匹配（如内置的输入法）

另外：对于测试还需要注意以下几点：

1. 并发（中断）测试：闹铃弹出框提示，另一个应用的启动、视频音频的播放，来电、用户正在输入等，语音、录音等的播放时强制其他正在播放的要暂停；

2. 数据来源的测试：输入，选择、复制、语音输入，安装不同输入法输入等；

3. push（推送）测试：在关机、待机状态下执行推送，消息先死及其推送跳转的正确性；应用在开发、未打开状态、应用启动且在后台运行的情况下是 push 显示和跳转否正确；推送消息阅读前后数字的变化是否正确；多条推送的合集的显示和跳转是否正确；

4. 分享跳转：分享后的文案是否正确；分享后跳转是否正确，显示的消息来源是否正确；

5. 触屏测试：同时触摸不同的位置或者同时进行不同操作，查看客户端的处理情况，是否会 crash 等

@软件测试总结及敏捷测试的流程：

<http://www.cnblogs.com/fnng/archive/2012/08/04/2622463.html>