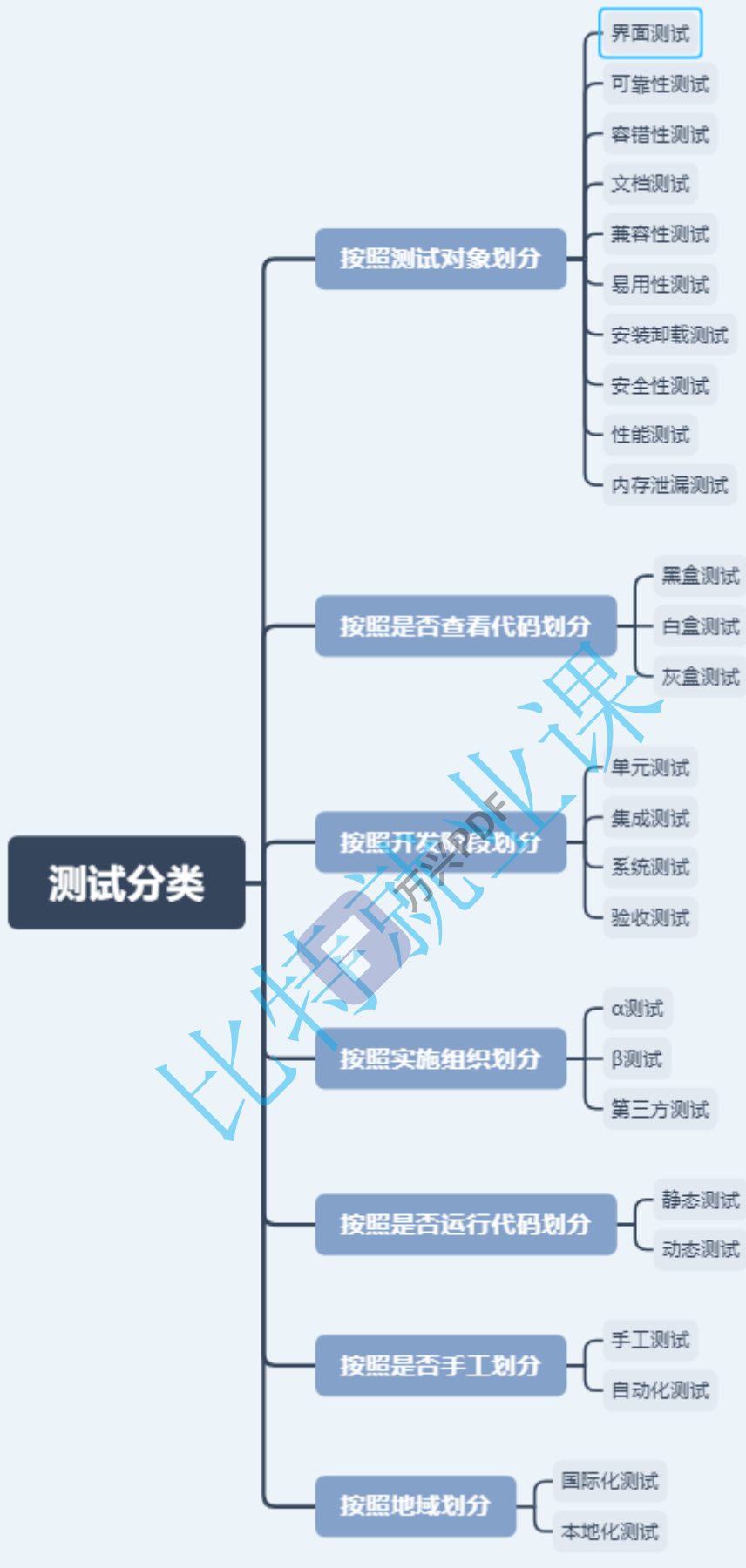




软件测试教程 第五节 进阶篇

本课程主要讲解软件测试的各种技术。作为一个测试人员，需要不断扩充自己的知识，并将各种知识用于项目测试中。在这里我们根据以上的测试技能图来讲解

万兴PDF
比兴PDF



按测试对象划分

界面测试

软件只是一种工具，软件与人的信息交流是通过界面来进行的，界面是软件与用户交流的窗口。界面是软件的第一层，界面的设计决定了用户对我们设计的软件的第一印象；界面如同人的面孔，具有吸引用户的直接优势，设计合理的界面能给用户带来轻松愉悦的感受。

界面测试（简称UI测试），指按照界面的需求（一般是UI设计稿）和界面的设计规则，对我们软件界面所展示的全部内容进行测试和检查，一般包括如下内容：

- 验证界面内容显示的完整性，一致性，准确性，友好性。比如界面内容对屏幕大小的自适应，换行，内容是否全部清晰展示；
- 验证整个界面布局和排版是否合理，不同板块字体的设计，图片的展示是否符合需求；
- 对界面不同控件的测试，比如，对话框，文本框，滚动条，选项按钮等是否可以正常使用，有效和无效的状态是否设计合理；
- 界面的布局和色调符合当下时事的发展。

思考：常见的界面错误有哪些？
重叠，截断，文字不合理自动换行等。

可靠性测试

可靠性（Availability）即可用性，是指系统正常运行的能力或者程度，一般用正常向用户提供软件服务的时间占总时间的百分比表示。

可靠性 = 正常运行时间 / (正常运行时间 + 非正常运行时间) * 100%

系统非正常运行的时间可能是由于硬件，软件，网络故障或任何其他因素（如断电）造成的，这些因素能让系统停止工作，或者连接中断不能被访问，或者性能急剧降低导致不能使用软件现有的服务等。

可用性指标一般要求达到4个或5个“9”，即99.99%或者99.999%

如果可用性达到99.99%，对于一个全年不间断（7*24的方式）运行的系统，意味着全年（252600min）不能正常工作的时间只有52min，不到一个小时。
如果可用性达到99.999%，意味着全年不能正常工作的时间只有5min。

不同的应用系统，可用性的要求是不一样的，非实时性的信息系统或一般网站要求都很低，99%和99.5%就可以了，但是军事系统，要求则很高；

课后思考：
如何进行可用性的测试？

容错性测试

容错性测试是指系统能够处理异常，用户的错误操作而不至于系统崩溃，从而提高系统的可用性。

容错性测试包含以下方面：

- 输入异常数据或进行异常操作，以检验系统的保护性。如果系统的容错性好，系统只给出提示或内部消化掉，而不会导致系统出错甚至崩溃。
比如数据级测试，校验测试，环境容错性测试，界面容错性测试
- 灾难恢复性测试。
通过各种手段，让软件强制性地发生故障，然后验证系统已保存的用户数据是否丢失，系统和数据是否能尽快恢复。

文档测试

国家有关计算机软件产品开发文件编制指南中共有14种文件，可分为3大类。

-开发文件：可行性研究报告、软件需求说明书、数据要求说明书、概要设计说明书、详细设计说明书、数据库设计说明书、模块开发卷宗。

-用户文件：用户手册、操作手册，用户文档的作用：改善易安装性；改善软件的易学性与易用性；改善软件可靠性；降低技术支持成本。

-管理文件：项目开发计划、测试计划、测试分析报告、开发进度月报、项目开发总结报告。

在实际的测试中，最常见的是用户文件的测试，例如：手册说明书等。也会有一些公司对需求文档进行测试，来保证需求文档的质量。

文档测试的关注点：

- 文档的术语
- 文档的正确性
- 文档的完整性
- 文档的一致性
- 文档的易用性

兼容性测试

大家经常上网，同一网站在不同的浏览器上表现不一样，有遇到过吗？IE-工具-兼容视图设置

兼容性测试需求是指明确要测试的兼容环境，考虑软，硬件的兼容，就软件兼容来说，主要考虑以下几个方面：

- 系统自身版本的兼容，用户已有数据的兼容，数据兼容是重中之重，对用户来说，数据是最有价值的。
- 测试与应用环境的兼容性，比如操作系统，应用平台，浏览器的兼容
- 测试与第三方系统以及第三方数据的兼容性

对于环境（操作系统，应用平台）兼容性的测试不仅仅局限在操作系统，浏览器这两个因素，还包括以下，32位，64位CPU；手机平台Android，iOS，Windows Phone；支持不同的Internet连接速度。

对于iOS和Android两个平台，还要区分手机和平板电脑，考虑不同的型号（屏幕尺寸，分辨率等）。

易用性测试

许多产品都应用人体工程学的研究成果，是产品在使用起来更加灵活和，舒适。软件产品也始终关注用户体验，让用户获得舒适，易用的体验，针对软件这方面的测试称之为易用性测试。

易用性在ISO25020标准中指容易发现，容易学习和容易使用。易用性包含七个要素：符合标准和规范，直观性，一致性，灵活性，舒适性，正确性和实用性。我们主要讨论以下几个方面

1，标准性和规范性

对于现有的软件运行平台，通常其UI标准已经不知不觉地被确立了，成为大家的共识。多数用户已经习惯并且接受了这些标准和规范，或者说已经认同了这些信息所代表的含义。比如安装软件的界面的外观，在什么场合使用恰当的对话框等。

所以用户界面上的各中信息应该符合规范和习惯，否则用户使用起来会不舒适，并得不到用户的认可。

测试人员需要把与标准规范，习惯不一致的问题报告为缺陷

2，直观性

用户界面的直观性，要求软件功能特性易懂，清晰。用户界面布局合理，对操作的响应在用户的预期之中。比如数据统计结果用报表的形式（条形图，扇形图等）展示清晰直观；现在主流的很多搜索引擎和日历的设计也有直观性的特点；



3, 灵活性

软件可以有不同的选项以满足不同使用习惯的用户来完成相同的功能。但是灵活性的设计要把握好度,不然可能由于太多的用户状态和方式的选择,增加了软件设计的复杂性,和程序实现的难度。

例如手机键盘有九宫格和全键盘,还支持手写,满足了不同用户的需求

4, 舒适性

舒适性主要强调界面友好, 美观, 操作过程顺畅, 色彩运用恰当, 按钮的立体感等。例如左手鼠标的设置给习惯用左手的人带来了便利, 也为右手十分劳累时提供了另一种途径;

安装卸载测试

应用的安装和卸载在任何一款APP中都属于最基本功能。一旦出错, 就属于优先级为紧要Critical的缺陷。主要需要考虑以下方面:

- 软件不同的安装和卸载方式;
- 应用是否可以在不同的环系统, 版本下安装 (安装兼容性)
- 安装或者卸载过程中是否可以手动暂停, 或者取消
- 安装空不足的时候系统是否有提示
- 是否可以正常的卸载, 以及应用软件的各种卸载方式
- 卸载和安装过程中出现环境问题, 软件是否可以正常并且合理的应对, 比如死机, 断电, 断网等

安全测试

安全性是指信息安全, 是指计算机系统或网络保护用户数据隐私, 完整, 保护数据正常传输和抵御黑客, 病毒攻击的能力。安全性测试属于非功能性测试很重要的一个方面, 系统常见的安全漏洞和威胁如下

- 输入域, 如输入恶性或者带有病毒的脚本或长字符串;
- 代码中的安全性问题, 如SQL/XML注入
- 不安全的数据存储或者传递
- 数据文件, 邮件文件, 系统配置文件等里面有危害系统的信息或者数据;
- 有问题的访问控制, 权限分配等
- 假冒ID: 身份欺骗
- 篡改, 对数据的恶意修改, 破坏数据的完整性

安全性测试的方法有代码评审, 渗透测试, 安全运维等, 常用的静态安全测试工具有, Coverity, IBM Appscan Source, HPFortify, 常用的动态安全测试有OWASP的ZAP, HP WebInspect等。其中静态安全测试是常用的安全性测试的方法。

4问题:

对于上传和下载的安全性该如何测试?

性能测试

我们在使用软件的时候有时会碰到软件网页打开时越来越慢, 查询数据时很长时间才显示列表, 软件运行越来越慢等问题, 这些问题都是系统的性能问题引起的。

要进行软件产品的性能问题, 要对产品的性能需求进行分析, 然后基于系统的性能需求和系统架构, 完成性能测试的设计和执行, 最后要进行持续的性能调优。常见的性能问题如下:

- 资源泄露
- 资源瓶颈
- 线程死锁, 线程阻塞
- 查询速度慢或效率低
- 受外部系统影响越来越大

衡量一个系统性能好坏的关键性指标有, 用户响时间, 事务平均响应时间 (TPS), 吞吐率, 每秒点击次数, 内存和CPU使用率等。

内存泄漏测试

很多软件系统都存在内存泄露的问题，尤其是缺乏自动垃圾回收机制的“非托管”语言编写的程序，例如C、C++、Delphi等。从用户使用的角度来看，内存泄露本身不会造成什么危害，一般用户可能根本不会感觉到内存泄露的存在。但是内存泄露是会累积的，只要执行的次数足够多，最终会耗尽所有可用内存，使软件的执行越来越慢，最后停止响应。可以把这种软件的问题比喻成软件的“慢性病”。

造成内存泄露的原因有很多，最常见的有以下几种。

- 分配完内存之后忘了回收。
- 程序写法有问题，造成没办法回收（如死循环造成无法执行到回收步骤）。
- 某些API函数的使用不正确，造成内存泄露。

内存泄漏的检测方法

- 人工静态法：代码走读，人工查找未被回收的内存。
- 自动工具法：借助相应测试内存泄漏的工具，如Visual Leak Detector，记录每次内存分配，清楚告诉用户内存是如何泄漏的。

按是否查看代码划分

在软件测试岗位的面试中，常常被面试官问到的概念就是黑盒和白盒的测试了，下面就来彻底讨论一下

黑盒测试(Black-box Testing)

黑盒测试就是在完全不考虑程序逻辑和内部结构的情况下，检查系统功能是否按照需求规格说明书的规定正常使用、是否能适当的接收输入数据而输出正确的结果，满足规范要求。

所以，黑盒测试又称之为数据驱动测试，只注重软件的功能。

黑盒测试的优点

- 不需要了解程序内部的代码以及实现，不关注软件内部的实现。
- 从用户角度出发设计测试用例，很容易知道用户会用到哪些功能，会遇到哪些问题，锻炼测试人员的产品思维
- 测试用例是基于软件需求开发文档，不容易遗漏软件需求文档中需要测试的功能。

黑盒测试的缺点是不可能覆盖所有代码。

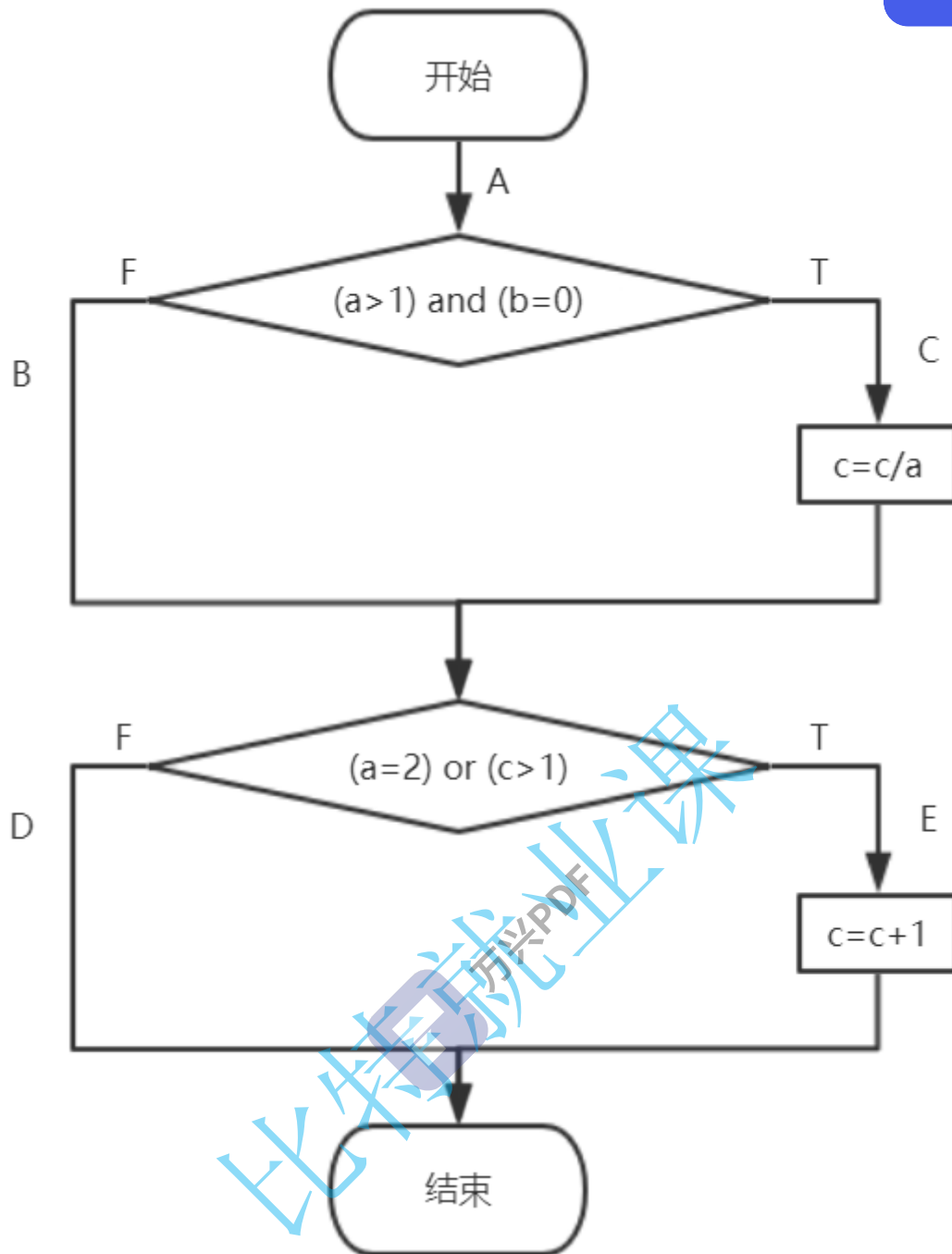
黑盒测试用到的测试方法有，等价类，边界值，因果图，场景法，错误猜测法等。

白盒测试(White-box Testing)

白盒测试又称为结构测试或逻辑测试，它一般用来分析程序的内部结构，针对程序的逻辑结构来设计测试用例进行测试。

白盒测试的测试目的是，通过检查软件内部的逻辑结构，对软件中的逻辑路径进行覆盖测试；在程序不同地方设立检查点，检查程序的状态，以确定实际运行状态与预期状态是否一致。

主要包含六种测试方法：**语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。**

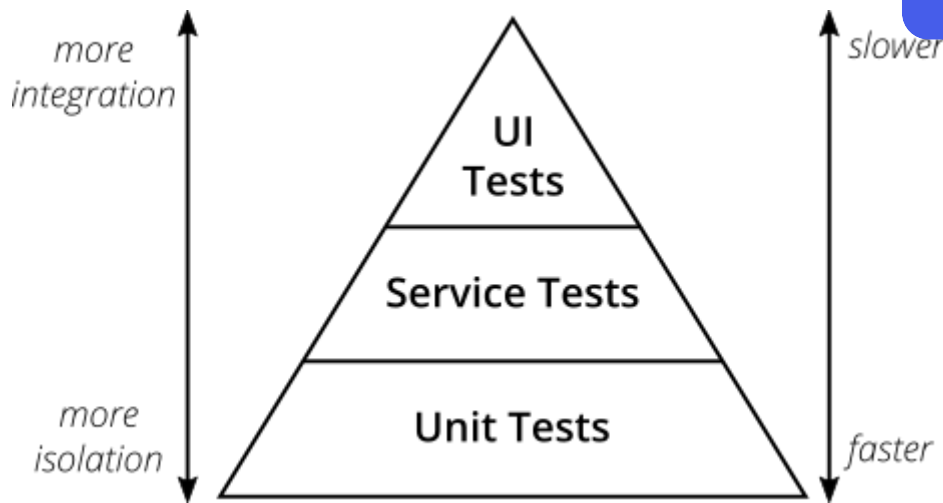


灰盒测试(Gray-Box Testing)

灰盒测试，是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段，不仅关注输出、输入的正确性，同时也关程序内部的情况。

按开发阶段划分

测试金字塔



单元测试(Unit Testing)

单元测试是对软件组成单元进行测试。其目的是检验软件基本组成单位的正确性。测试的对象是软件设计的最小单位：模块。又称为模块测试

测试阶段：编码后或者编码前（TDD）

测试对象：最小模块

测试人员：白盒测试工程师或开发工程师

测试依据：代码和注释+详细设计文档

测试方法：白盒测试

测试内容：模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试

Java如何进行单元测试？

集成测试(Integration Testing)

集成测试也称联合测试（联调）、组装测试，将程序模块采用适当的集成策略组装起来，对系统的接口及集成后的功能进行正确性检测的测试工作。集成主要目的是检查软件单位之间的接口是否正确。

- 测试阶段：一般单元测试之后进行
- 测试对象：模块间的接口
- 测试人员：白盒测试工程师或开发工程师
- 测试依据：单元测试的模块+概要设计文档
- 测试方法：黑盒测试与白盒测试相结合
- 测试内容：模块之间数据传输、模块之间功能冲突、模块组装功能正确性、全局数据结构、单模块缺陷对系统的影响

系统测试(System Testing)

新买手机都会有一个合格标签，在出厂前手机厂会对其型号的手机上的所有功能全部测试一遍。包括手机硬件本身，手机上自带的APP。

将软件系统看成是一个系统的测试。包括对功能、性能以及软件所运行的软硬件环境进行测试。

- 测试阶段：集成测试通过之后
- 测试对象：整个系统（软、硬件）

- 测试人员：黑盒测试工程师
- 测试依据：需求规格说明文档
- 测试方法：黑盒测试
- 测试内容：功能、界面、可靠性、易用性、性能、兼容性、安全性等

回归测试(Regression Testing)

接着上边的美颜功能不可用的例子，拿去维修点进行了维修，拿到手机后第一件事情是先看美颜功能修好了没有，第二件事情就是看看手机的其它常用功能是否正常。

回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

在整个软件测试过程中占有很大的工作量比重，软件开发的各个阶段都会进行多次回归测试。随着系统的庞大，回归测试的成本越来越大，通过选择正确的回归测试策略来改进回归测试的效率和有效性是很有意义的。

冒烟测试 (smoke testing)

冒烟测试的对象是每一个新编译的需要正式测试的软件版本，目的是确认软件主要功能和核心流程正常，在正式进行系统测试之前执行。冒烟测试一般在开发人员开发完毕后提交给测试人员来进行测试时，先进行冒烟测试，保证基本功能正常，不阻碍后续的测试。

如果冒烟测试通过，则测试人员开始进行正式的系统测试，如果不通过，则测试人员可以让开发人员重新修复代码直到冒烟测试通过，再开始进行系统测试。

回归测试和冒烟测试都属于系统测试。

验收测试(Acceptance Testing)

买到新手机，一般会有7天包退，一个月包换，我们会尽量在7天内把手机的所有功能都试一遍。

验收测试是部署软件之前的最后一个测试操作。它是技术测试的最后一个阶段，也称为交付测试。验收测试的目的是确保软件准备就绪，按照项目合同、任务书、双方约定的验收依据文档，向软件购买都展示该软件系统满足原始需求。

- 测试阶段：系统测试通过之后
- 测试对象：整个系统（包括软硬件）。
- 测试人员：主要是最终用户或者需求方。
- 测试依据：用户需求、验收标准
- 测试方法：黑盒测试
- 测试内容：同系统测试(功能...各类文档等)

按测试实施组织

α测试(Alpha Testing)

手机出厂前最后一次测试，开发和测试人员不参与。

α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。α测试的目的是评价软件产品的FLURPS(即功能、局域化、可使用性、可靠性、性能和支持)。

大型通用软件，在正式发布前，通常需要执行Alpha和Beta测试。α测试不能由程序员或测试员完成。

β测试(Beta Testing)

新手机购买回来，参与测试的人是购买者，使用的场所及环境已不再是手面厂商的环境及场所。

Beta测试是一种验收测试。Beta测试由软件的最终用户们在一个或多个场所进行。

α测试与Beta测试的区别：

测试的场所不同：Alpha测试是指把用户请到开发方的场所来测试,beta测试是指在一个或多个用户的场所进行的测试。

Alpha测试的环境是受开发方控制的,用户的数量相对比较少,时间比较集中。beta测试的环境是不受开发方控制的,用户数量相对比较多,时间不集中。

alpha测试先于beta测试执行。通用的软件产品需要较大规模的beta测试,测试周期比较长。

第三方测试

介于开发方和用户方间的组织的测试。

按是否运行划分

静态测试(Static testing)

所谓静态测试 (static testing) 就是**不实际运行被测软件**,而只是**静态地检查程序代码、界面或文档**中可能存在的错误的过程。不以测试数据的执行而是**对测试对象的分析过程**,仅通过分析或检查源程序的设计、内部结构、逻辑、代码风格和规格等来检查程序的正确性。

动态测试(Dynamic testing)

动态测试 (dynamic testing), 指的是**实际运行被测程序**,**输入相应的测试数据**,**检查实际输出结果和预期结果是否相符**的过程,所以判断一个测试属于动态测试还是静态的,唯一的标准就是看是否运行程序。

大多数软件测试工作都属于动态测试。

按是否手工划分

手工测试(Manual testing)

手工测试就是由人去一个一个的输入用例,然后观察结果,和机器测试相对应,属于比较原始但是必须的一个步骤。总结优缺点:

- 优点: 自动化无法替代探索性测试、发散思维结果的测试。
- 缺点: 执行效率慢,量大易错。

自动化测试(Automation Testing)

就是在预设条件下运行系统或应用程序,评估运行结果,预先条件应包括正常条件和异常条件。简单说自动化测试是把以人为驱动测试行为转化为机器执行的一种过程。

自动化测试比如功能测试自动化、性能测试自动化、安全测试自动化。

自动化测试按照测试对象来分,还可以分为接口测试、UI测试等。接口测试的ROI(产出投入比)要比UI测试高。

自动化实施步骤:

1. 完成功能测试,版本基本稳定
2. 根据项目特性,选择适合项目的自动化工具,并搭建环境
3. 提取手工测试的测试用例转化为自动化测试的用例
4. 通过工具、代码实现自动化的构造输入,自动检测输出结果是否符合预期
5. 生成自动测试报告
6. 持续改进,脚本优化。

按测试地域划分

什么是软件国际化？

是在软件设计和文档开发过程中，使得功能和代码设计能处理多种语言和文化习俗，使创建不同语言版本时，不需要重新设计源程序代码的软件工程方法。

国际化测试

软件的国际化 and 软件的本地化是开发面向全球不同地区用户使用的软件系统的两个过程。而本地化测试和国际化测试则是针对这类软件产品进行的测试。由于软件的全球化普及，还有软件外包行业的兴起，软件的本地化和国际化测试俨然成为了一个独特的测试专门领域。

本地化和国际化测试与其他类型的测试存在很多不同之处。下面是本地化和国际化测试的一些要点。

- 1、本地化后的软件在外观上与原来版本是否存在很大的差异，外观是否整齐、不走样。
- 2、是否对所有界面元素都进行了本地化处理，包括对话框、菜单、工具栏、状态栏、提示信息（包括声音的提示）、日志等。
- 3、在不同的屏幕分辨率下界面是否正常显示。
- 4、是否存在不同的字体大小，字体设置是否恰当。
- 5、日期、数字格式、货币等是否能适应不同国家的文化习俗。例如，中文是年月日，而英文是月日年。
- 6、排序的方式是否考虑了不同语言的特点。例如，中文按照第一个字的汉语拼音顺序排序，而英文按照首字母排序。
- 7、在不同的国家采用不同的度量单位，软件是否能自适应和转换。
- 8、软件是否能在不同类型的硬件上正常运行，特别是在当地市场上销售的流行硬件上。
- 9、软件是否能在Windows或者其他操作系统的当地版本上正常运行。
- 10、联机帮助和文档是否已经翻译，翻译后的链接是否正常。正文翻译是否正确、恰当，是否有语法错误。

软件本地化和国际化测试是一个综合了翻译行业和软件测试行业的测试类型。它要求测试人员具备一定的翻译能力、语言文化，同时具备测试人员的基本技能。

本地化测试

之前我们所讲的全是本地化测试。