**小白入门专用—软件测试基础知识**

先前自己网上整理的一些资料——其中包括测试定义、测试方法分类、测试原则、测试策略、测试模型，希望对和我一样刚刚入门的小伙伴有些许帮助  理论知识还是基础，重点还是要在项目实战里运用贯通知识点。

**1.定义（目的）：发现软件程序中的错误而执行程序的过程**

1）软件测试为了发现程序存在的代码或业务逻辑错误；2）软件测试为了检验产品是否符合用户需求；3）软件测试为了提高用户的体验

**2.测试方法分类（能按照不同的维度区分软件测试方法的分类，熟悉每一个方法的特点）：**

**按开发阶段分**

单元测试：对软件组成单元进行测试，目的是验证软件基本组成单位的正确性，测试对象是软件设计的最小单位：模块/ 集成测试：也称联合测试、组装测试，将程序模块采用适当的集成策略组装起来，对系统接口集成后的功能进行检测，验证软件单位间接口是否正确。

系统测试：就是一个系统的测试。包括功能、性能、软件运行软硬件环境的进行测试，大部分时间是在这个阶段。

验收测试：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，展示系统的原始需求是否满足。

**按是否运行**

静态测试：指不运行被测程序，仅仅通过分析或者检查源程序的语法、结构、过程、接口等来检查程序的正确性。

对需求规格说明书、软件设计说明书、对源程序做结构分析、流程图分析、符号执行来找bug。

动态测试：通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率、正确性、健壮性。这种方法通过三部分组成：构造测试用例、执行程序、分析程序的输出结果。

**按是否查看代码**

白盒测试：结构测试、透明测试、逻辑驱动测试、基于代码测试，即研究源代码和程序的结果

黑盒测试：只关心软件的输入输出

**功能测试**

功能测试：测试软件功能是否符合需求

界面测试：UI测试，系统界面是否合理，整体风格是否一致，界面文字是否正确，图片是否美观、冒烟测试：对象是每一个新编译的需要正式测试的软件版本，确认基本功能正常，后续可以正式测试工作

 回归测试：错误修正后或功能环境发生变化后进行的重新测试，确认修改不会影响其他功能

业务逻辑测试：侧重在业务流程，基本功能已合格的基础上，准备组合多种测试数据，来驱动或辅助各种约束条件下的业务流程，确定最终数据结果是否符合预期

兼容性测试：测试系统与其他软硬件的兼容性（APP、C/S架构、B/S架构）

易用性测试：测试软件在使用是用户是否使用觉得方便、主观性较强

**性能测试**

性能测试：为获取验证系统的性能指标而进行的测试，一般会在不同的负载情况进行

负载测试：通过改变系统负载的方式，增加负载来发现系统性能存在的问题

压力测试：强度测试，主要是确定系统的稳定性，高负载长时间稳定性压力测试，极限负载情况下导致系统崩溃的破坏性压力测试

容量测试：确定系统可以处理同时在线用户数的最大数量，使系统承受超额的数据容量来发现它是否能够正确处理

并发测试：测试多用户并发访问同一个应用，模块，数据时是否产生隐藏的并发问题，如内存泄漏，线程锁，资源争用，而几乎所有的性能测试都会用到并发测试

配置测试：对被测系统的软硬件环境调整，了解不同环境对系统性能影响程度，从而找到系统各项资源的最优分配原则

可靠性测试：评估产品在规定的寿命期限内，预期使用，存储，运输等所有环境，保持功能可靠性而进行活动

灰盒测试：介于白盒黑盒之间的测试，多用于集成测试阶段，不仅关注输入输出的正确性，同时也关注程序内部的情况。

**是否手工执行**

手工测试：手工的一个个去输入用例，然后观察结果，和机器测试相对应

自动化测试（Automation Testing）：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和异常条件，    就是把人为驱动的测试行为转化为机器执行的过程。

**其他**

冒烟测试：是在将代码更改嵌入到产品的源树中之前对这些更改进行验证的过程，冒烟测试设计用于确认代码中的更改会按预期运行，且不会破坏整个版本的稳定性

安全测试：测试系统防范非法入侵的能力

探索性测试：测试思维技术，没有很多实际测试方法，技术，工具，是测试人员应该具有    的一种能力，探索性测试强调人员的主观能动性，能够摒弃复杂的测试计划，测试用例设计过程，强调在碰到问题时及时改变测试策略。

回归测试：错误修正后，软件功能、环境发生变化后的重新测试，确认修改不会对其他功能造成影响。

Alpha测试：前期用户测试，公司内部员工与部分用户模拟实际操作环境下进行的验收测试，也称为内测

Beta测试：后期测试，系统已经通过内部测试大部分的错误已经修正即将正式发行，在一个、多个真实环境下发布版本，进行测试，也称为公测。

**3.软件测试原则（熟悉基本约定）：**

1）测试应该尽早介入；----需求分析2）所有的测试都应追溯到用户需求；3）程序员应该避免检查自己的程序。除了单元测试。因为程序员对于自己的作品，思维具有局限性。无法保证测试质量。交给第三方或者专业测试，运用各种测试技术，利用丰富的测试经验和对bug的敏感，去提高软件的质量；4）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，特殊情况下还要制造极端状态和意外状态。5）二八原则，测试发现的错误中80%很可能起源于20%的模块中；6）对错误结果要进行一个确认过程；7）制定严格的测试计划；8）完全测试是不可能的，测试需要终止；9）妥善保存测试过程中的所有文档

**4.软件测试策略（高效合理的安排测试任务）：**

测试策略在结构上可以包括：（1）测试级别：常见的测试级别有单元测试，集成测试和系统测试。大部分的测试组织里面，单元测试由开发负责，而集成测试和系统测试由测试部门或者质量保证部门负责。（2）角色与职责：需要在测试策略里面明确定义各个角色，以及该角色的职责。比如项目经理，测试组长，测试工程师…（3）环境需求：这一点非常重要，它将描述测试时需要的系统环境，包括软硬件以及网络环境等等。在澄清环境需求的时候，测试组织可以识别出资源方面的风险。（4）风险分析：影响测试过程的风险都应该尽早被识别出来，而且必须有相应的解决办法以便消除或者减轻这些风险。（5）测试进度：测试进度将会评估完成测试所需要的时间。在设定进度的时候，首先需要明确测试范围，然后根据测试资源的多少来制定能被各方面认可的测试进度计划。做一个非常准确的进度计划是困难的事情，因为测试过程中充满了各种不确定性，所以一般计划者需要考虑增加一定的buffer。当然，制定进度计划的时候可以参考已有的项目的数据。如果是一个全新的软件项目，专家认为将初始计划的时间翻倍比较靠谱！（6）回归测试方法：回归测试用来保证之前fix bug的代码不会影响软件的其他部分，这样需要我们选择已经执行过的测试用例重新运行。测试人员需要找到一个方法来确定哪些测试用例应该在回归测试中运行，用例不能太多，因为资源有限，用例也不能太少，否则会达不到必须的测试强度。不过，如果测试部门对待测系统以及软件架构非常了解的话，就比较容易找到合适的回归测试集合。（7）测试范围：要测试的内容，可能是某些模块，可能是某些指标，比如功能，性能，易用性…（8）测试优先级：测试范围内的东西不会都是一样重要的，加上测试资源各种有限，所以为测试排定优先级是十分的必要。

**5.软件测试模型（了解软件测试在项目研发过程中的定位）：**

**V模型**

V模型（测试 ）

1、单元测试又称模块测诚，针对软件设计中的最小单位—程庄模块，进行正确性检查的测试工作。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。单元定义：C中指一个函数，Java中指一个类，在图形化的软件中，单元一般指1个窗口，1个菜单。

2、集成测试又叫组装测试，通常在单元测试的基础上，将所有程序模块进行有序的、递增的测试。重点测试不同模块的接口部分。

3、系统测试(system testing)：指的是将整个软件系统看为一个整体进行测试，包括对功能、性能、以及软件所运行的软硬件环境进行测试。系统测试在系统集成完毕后进行测试，前期主要测试系统的功能是否满足需求，后期主要测试系统运行的性能是否满足需求，以及系统在不同的软硬件环境中的兼容性等。

4、验收测试α测试：Alpha是内测版本，即现在所说的C8，比版本表示该软件仅仅是一个初步完成品，通常只在软件开发者内部交流，也有很少一部分发布给专业测试人员。一般而言，该版本软件的bug较多，普通用户最好不要安装。

β测试：Beta是公测版本，是对所有用户开放的测试版本。该版本相对于a颜已有了很大的改进，消除了严重的错误，但还是存在着一些缺陷需要经过大规模的发布试来进一步消除。这一版本通常由软件公司免费发布，用户可从相关的站点下载。通过一些专业爱好者的测试，将结果反馈给开发者，开发者们再进行有针对性的修改。该版本也不适合一般用户安装。

λ测试：Camma版本，指的是软件版本正式发行的候选版。该版本已经相当成熟了，与即将发行的正式版相差无几，成为正式发布的候选版本。软件正式版本推出之前的几个版本，需要有人测试一下，看看是不是有问题。在开发该软件的公司内部的由该公司内部人员式的称为：Alpha测试，Alpha 测式主要看有没有功能缺失或系统错误，Alpha 测试完后一般不会有大问题了。然后把软件拿给用户测试称为：beta 测试，主要是看用户对软件外观、使用方便等的反应。这么多的测试版一方面为了最终产品尽可能地满足用户的需要，另一方面也尽量成少了软件中的bug。然后做过一些修改，成为正式发布的候选版本时，叫做gamma（现在叫做RC-ReleaseCandidate)。简单来说，阿尔法测试主要是测试人员在开发环境下的测试，贝塔测试是在实际环境中的测试，或者公司内部人员在模拟真实环境中的测试。

V模型的优缺点（测试重点）

1、优点：包含了底层测试（单元测试）和高层测试（系统测试）；清楚的标识了开发和测试的各个阶段；自上而下逐步求精，每个阶段分工明确，便于整体项目的把控。

2、缺点：自上而下的顺序导致了，测试工作在编码之后，就导致错误不能及时的进行修改；实际工作中，需求经常变化，导致v模型步骤，反复执行，返工量很大，灵活度较低。改良：每个步骤都可以进行小的迭代工作

**W模型（重要）**

定义：开发一个v；测试一个v也叫双V模型

优点： 开发伴随着整个开发周期，需求和设计同样要测试； 更早的介入测试，可以发现初期的缺陷，修复成本低； 分阶段工作，方便项目整体管理。缺点： 开发和测试依然是线性的关系，需求的变更和调整，依然不方便； 如果没有文档，根本无法执行w模型；对于项目组成员的技术要求更高！

**H模型**

 H模型的优点：

开发的H模型揭示了软件测试除测试执行外，还有很多工作；软件测试完全独立，贯穿整个生命周期，且与其他流程并发进行；软件测试活动可以尽早准备、尽早执行，具有很强的灵活性；软件测试可以根据被测物的不同而分层次、分阶段、分次序的执行，同时也是可以被迭代的。H模型的缺点：管理型要求高：由于模型很灵活，必须要定义清晰的规则和管理制度，否则测试过程将非常难以管理和控制；技能要求高：H模型要求能够很好的定义每个迭代的规模，不能太大也不能太小；测试就绪点分析困难：测试很多时候，你并不知道测试准备到什么时候是合适的，就绪点在哪里，就绪点的标准是什么，这就对后续的测试执行的启动带来很大困难；对于整个项目组的人员要求非常高：在很好的规范制度下，大家都能高效的工作，否则容易混乱。例如：你分了一个小的迭代，但是因为人员技能不足，使得无法有效完成，那么整个项目就会受到很大的干扰。