

一名合格测试的必知必会

前言

作为一名合格的测试，一方面应该具备基本的测试业务知识，另一方面也要有扎实的技术基础。测试是很有技术含量的工作，绝对不是一味的对界面点点点的重复操作。所以合格的测试一定是理论知识与技术能力并行发展。下面就给大家介绍下测试到底都应该掌握什么。

基本的测试理论

我这里不想给大家说一些可以很容易就在网上找到的东西，其实测试许多的理论应该是基于你对业务系统的流程、业务场景的基础上进行总结的。这才是真正的干货。

立足业务场景

测试任何功能点，不能单纯的看需求、设计文档就盲目的开始测试，无论是编写测试用例、还是执行测试用例都一定要把业务场景放在首位。即常说的“User Story”（用户故事），我们一定要模拟、分析出这条测试用例、这个测试思路是基于用户的哪种操作，并且要把自己定位于那种计算机盲，然后站在他们的角度去使用被测系统。举个例子：

本来正确的系统操作流程应该是先A后B在C，那我们就应该模拟先B后C在A、先C后A在B、先B后A在C等等非正常流程操作下的系统响应。顺便说一下，这里我设计用例的思维方式并不是凭空而来，而是排列组合思维。

注意历史数据

这个主要只我们对正在维护的系统，每次有改动点，一定要考虑其流程、改动对历史数据的影响。中国有句老话：“打江山容易，守江山难”，映射到测试工作上也是对于新功能的测试一般都很少出问题，往往对于历史数据，或者新功能点对老功能点的影响没有考虑到，才会导致BUG的产生。所以作为一名合格的测试，测试任何一个优化功能，一定要拿历史数据和新数据一同进行测试。这里也引申出我的第二个思维模式，对比思维。

极端情况

另外，测试任何一个功能点，我们一定要考虑到极端情况。例如：



QQ登录首页，我们谁都不陌生，假设你作为一名腾讯的登录首页测试人员，此时叫你测试【注册账号】这个功能，我们常规思维肯定会考虑以下情况：

1. 录入的长度是否有限制；
2. 容错性，是否支持特殊符号；
3. 兼容性，各个浏览器下是否都可以正常使用；
4. 跨平台性。APP、平板、电脑、乃至智能电视机是不是都支持；
5. 安全性，是否有SQL注入风险。

此时我们肯定会认为自己设计的用例涵盖了其主要功能，这个时候我们就漏掉了一种极端情况，即客户上来之后不录入任何情况，就点击【登录】按钮。遇到这种极端问题特别多的测试同仁肯定有感受，这种极端情况下暴露的BUG往往是最为致命、最为常见的“空指针异常”。综合类似上述的场景，引出我们合格的测试人必须要有的第三个思维模式，逆向思维。

MySQL语句的书写能力

SQL语句其实在IT行业里比较难的技术，如果你不认同，那说明你接触的还比较浅薄。当然也绝不是只掌握了简单的增、删、改、查就可以的。那么我们应该掌握哪些方面呢？

1. 第一组：基本语句的书写，包括：`select from`（查询）、`delete from`（删除）、`where`（条件子句）、`in`（在...里）、`not in`（不在...里）、`from`（从）、`between ...and`、`update(更新)..set`、`Alter`（修改表、列）、`insert`（插入）..`values`；
2. 第二组：进阶语句的书写：`group ..by`（分组）、`count(*)`与`count(1)`、`having`（分组条件语句）、`left ..join on`（左连接）、`right(右连接)`、等值连接。
3. 函数学习：`sum`等数学函数，还有一个必须要掌握的`case...when..then...end`。

除了上述的基本语句的书写外，还应该掌握：

1. 视图的建立。
2. 存储过程的书写，会存储过程，可以方便我们造数。

介于SQL这部分的知识在网上都可以找到，不在详述。

Linux 基本命令的掌握

这里我只点出具体我们测试人员需要掌握的常用的linux命令，没有提到的，工作中你90%用不到。有兴趣可以自行深入研究下。

1. ll与ls（查看当前文件）、cd（切换到指定文件夹）、pwd（查看当前路径）。
2. tail -f 日志名称（查看后台日志的时候经常使用）。
3. ps -ef | grep java。查看当前tomcat下服务器进程的时候使用。
4. kill -9 进程号。杀死当前的指定进程。
5. view 显示正在运行的文件信息。

除了上述的五组命令外，其他的命令基本用不到。

测试分析与测试用例

有些时候时间紧、测试任务重，这个时候你可以不用详细的书写测试用例，但是一定要有完备的测试分析。分析的内容应该包含以下方面。

从开发角度分析

1. 产品和特性是由哪个开发具体负责的,可以方便后续很好的跟踪问题。
2. 开发此产品和特性是否采用了新技术。
3. 自测过得流程有哪些？
4. 是否还有为解决的问题？
5. 认为的风险点在哪里？
6. 对测试的测试建议是什么？

从测试的角度分析

1. 基于的业务场景是什么？
2. 产品面向的人群是谁？
3. 测试工作如何计划？
4. 测试人员如何分配？
5. 测试完成的衡量目标是什么？
6. 采用的测试技术、工具应该有哪些？
7. 还有没有需要开发、需求支持的点？

8. 分析出测试重点、难点、分配测试资源。

上面说的分析过程也是我经常用到的68原则，即充分了解产品，并且认真分析之后再去实施，肯定会使得你的测试达到事半功倍的效果。

测试用例的编写

这里我只说我写用例的一个小习惯，各个公司的测试用例书写规范不尽相同，但是请一定要把每一条测试用例执行时的“前提条件”标注清晰。这个“前提条件”可以是，这条用例执行前的：

1. 环境准备。
2. 业务流程前提，即做到哪一步，才满足执行这条用例的条件。
3. 数据库状态，此时的数据库相关表的状态、存储是怎样的。
4. 也可以是网络情况，是内网下还是外网下。
5. 还可以是介质，是手机、电脑、还是平板电脑。
6. 可以是系统，是windows下的执行测试用例还是linux系统下的测试用例。

等等。只有标清楚“前提条件”后期维护方便、并且可以叫人阅读时一目了然。

开发语言 GitChat

另外测试最好掌握一门开发语言，这个语言我只推荐两种Java或者python。会了开发语言，请记住你掌握的是技能，请不要转换自己的身份和思维，掌握开发语言就站在开发实现角度去思索软件。配合sonarqube的静态扫描，你会发现你会慢慢的深入代码层去解决测试问题。

沟通技巧

这个不多少了，记住两点：

1. 与任何项目组的人大家的出发点都是为了项目，所以要相互理解，学会换位思考；
2. 要坚持自己原则，不要輕易被开发的思维所诱导，而放弃自身的思维模式。

总结

测试的道路上其实充满挑战，做的好，你的价值一定远远高于开发、产品，你要记住你是每一个项目、每一个产品的质检员，你的能力高低直接关乎产品的质量、价值乃至口

碑，所以请从现在开始，加油充电，早早成为一名合格的测试！

GitChat