**01|你真的懂测试吗?从”用户登陆”测试谈起**

作为专栏的第一篇文章,我选择了一个你耳熟能详的”用户登录”功能为测试对象，希望通过这样一个简单直白的功能帮助你理解如何做好测试，以及现阶段你需要加强和提高的测试技能。

可能你会说，“用户登陆”这个测试对象也太简单了吧，我只要找一个用户让他在界面上输入用户名和密码，然后点击“确认”按钮，验证一下是否登录成功就可以了。的确，这构成了一个最基本、最典型的测试用例，这也是终端用户在使用系统时最典型的Happy Path场景。

但是做为测试工程师，你的目标是要保证系统在各种应用场景下的功能是否符合设计要求的，所以你需要考虑的测试用例就需要更多、更全面，于是你可能会根据“用户登录”功能的需求描述，结合等价类划分和边界值分析方法来设计一系列的测试用例。

那什么是等价划分和边界值分析方法呢？首先，这二者都隶属于最常用、最典型、也是最重要的黑盒测试方法。

* 等价类划分方法，是所有可能的输入数据划分成若干个子集，在每个子集中，如果任意一个输入数据对于揭露程序中潜在错误都具有同等效果，那么这样的子集就构成了一个等价类。后续只要从每个等价类中任意选取一个值进行测试，就可以用少量具有的测试输入取得较好的测试覆盖结果。
* 边界值分析方法，是选取输入、输出的边界值进行测试。因为通常大量的软件错误是发生在输入或输出范围的边界上，所以需要对边界值进行重点测试，通常选取正好等于、刚刚大于或刚刚小于边界的值作为测试数据。

从方法论上可以看出来，边界值分析是对等价类划分的补充，所以这两种测试方法经常结合起来使用。

现在，针对“用户登录”功能，基于等价划分和边界值分析方法，我们设计的测试用例包括：

1. 输入已注册的用户名和正确的密码，验证是否登录成功；
2. 输入已注册的用户名和不正确的密码验证是否登录失败，并且提示信息正确；
3. 输入末注册的用户名和任意密码，验证是否登录失败，并且提示信息正确；
4. 用户名和密码两者都为空，验证是否登录失败，并且提示信息正确；
5. 用户名和密码两者之一为空，验证是否登录失败，并且提示信息正确；
6. 如果登录功能启用了验证码功能，在用户名和密码正确的前提下，输入正确的验证码，验证是否登录成功。
7. 如果登录功能启用了验证码功能，在用户名和密码正确的前提下，输入错误的验证码，验证是否登录失败，并且提示信息正确。

列出这些测试用例后，你可能已经觉得比较满意了，因为你觉得已经把自己测试知道都用在这些用例设计中了。、

的确，上面的测试用例集已经涵盖了主要的功能测试场景。但是在一个优秀的测试工程师眼中，这些用例只能达到勉强及格的标准。

什么？才刚刚及格？如果你有这个想法，那我建议你在继续看下面的内容前，先仔细思考一下，这些测试用例是否真的还需要扩充。

现在，我跟你分享一下有经验的测试工程师会再增加的测试用例：

1. 用户名和密码是否大小写敏感；
2. 页面上的密码框是否加密显示；
3. 后台系统创建的用户第一次登录成功时，是否提示修改密码；
4. 忘记用户名和忘记密码的功能是否可用；
5. 前端页面是否根据设计要求限制用户名和密码长度；
6. 如果登录功能需要验证码，点击验证码图片是否可以更换验证码，更换后的验证码是否可用；
7. 刷新页面是否会刷新验证码；
8. 如果验证码具有时效性，需要分别验证时效内和时效外验证码的有效性；
9. 用户登录成功但是会话超时后，继续操作是否会重定向到用户登录界面；
10. 不同级别的用户，比如管理员用户和普通用户，登录系统后的权限是否正确；
11. 页面默认焦点是否定位在用户名的输入框中；
12. 快捷键Tab和Enter等，是否可以正常使用。

看完这些用例，你可能会说：“哇塞，原来一个简简单单的登录功能居然有这么多需要测试的点”。但是，你别高兴的得太早，“用户登录”功能 的测试还没有结束。

虽然改进后的测试用例集相比之前的测试覆盖率的确已经提高了很多，但是站在资深测试人员的角度来看，还有很多用例需要设计。

经我这么一说，你可能已经发现，上面所有的测试用例设计都是围绕显示功能性需求的验证展开的，换名话说，这些用例都是直接针对“用户登录”功能的功能性进行验证和测试。

但是，一个质量过硬的软件系统，除了显式功能性需求以外，其他的非功能性需求即隐式功能性需求也是极其关键的。

显式功能性需求（Functional requirement）的含义从字面上就可以很好的理解，批的是软件本身需要实现的具体功能，比如“正常用户使用正确的用户名和密码可以成功登录”、“非注册用户无法登录”等，这都属于典型的显式功能性需求描述。

那什么是非功能性需求（Non-functional requirement）呢？从软件测试的维度来看，非功能性需求主要涉及安全性、性能以及兼容性三大方面。在上面所有的测试用例设计中，我们完全没有考虑对非功能性需求的测试但这些往往是决定软件质量的关键因素。

明白了非功能性需求测试的重要性后，你可以先思考一下还需要设计哪些测试用例，然后再来看看我会给出哪些用例，相信这种方式对你的帮助会更大。

安全性测试用例包括：

1. 用户密码后台存储是否加密；
2. 用户密码在网络传输过程中是否加密；
3. 密码是否具有有效期，密码有效期到其后，是否提示需要修改密码；
4. 不登录的情况下，在浏览器中直接输入登录后的URL地址，验证是否会重新定向到用户登录界面；
5. 密码输入框是否不支持复制和粘贴；
6. 密码输入框内输入的密码是否都可以在页面源码模式下被查看；
7. 用户名和密码的输入框中分别输入典型的“SQL注入攻击”字符串，验证系统的返回页面；
8. 用户名和密码输入框中分别输入典型的“XSS跨站脚本攻击”字符串，验证系统行为是否被篡改；
9. 连续多次登录失败情况下，系统是否会阻止后续的尝试以应对暴力破解；
10. 同一用户在同一终端的多种浏览器上登录，验证登录功能的互斥性是否符合设计预期；
11. 同一用户先后在多台终端的浏览器上登录，验证登录是否具有互斥性。

性能压力测试用例包括：

1. 单用户登录的响应时间是否小于3秒；
2. 单用户登录时，后台请求数量是否过多 ；
3. 高并发场景下用户登录的响应时间是否小于5秒；
4. 高并发场景下服务端的监控指标是否符合预期；
5. 高集全点并发场景下，是否存在资源死锁和不全理的资源等待；
6. 长时间大量用户连续登录和登出，服务器端是否存在内在泄漏。

兼容性测试用例包括：

1. 不同浏览器下，验证登录页面的显示以及功能正确性；
2. 相同浏览器的不同版本下，验证登录页面的显示以及功能正确性；
3. 不同移动设备终端的不同浏览器下，验证登录页面的显示以及功能正确性；
4. 不同分辨率的界面下，验证登录页面的显示以及功能正确性。

说到这里，你还会觉得“用户登录”功能的测试非常简单、不值一提么？一个看似简单的功能测试，居然涵盖了如此多的测试用例，除了要覆盖明确的功能性需求，还需要考虑其他诸多的非功能性需求。

另外，通过这些测试用例的设计，你也可以发现，一个优秀的测试工程师必须具有很宽广的知识面，如果你不能对被测系统的设计有深入的理解、不明白安全攻击的基本原理、没有掌握性能测试的基本设计方法，很难设计出“有的放矢”的测试用例。

通过“用户登录”功能测试这个实例，我希望可以激发你对测试更多的思考，并且开拓你设计测试用例的思路，以达到抛砖引玉的效果。

看完了这些测试用例，你可能会说还有一些遗漏的测试点没有覆盖到，这个功能的测试点还不够全面。那么，接下来我再跟你说说测试点还不够全面。那么，接下来我再跟你说说测试的不可穷尽性，即绝大多数情况下，是不可能进行穷尽测试的。

所谓的“穷尽测试”是指包含了软件输入值和前提条件所有可能组合的测试方法，完成穷尽测试的系统里应该不残留任何未知的软件缺陷。因为如果有未知的软件缺陷，你可以通过做更多的测试来找到它们，也就是说你的测试还没有穷尽。

但是，在绝大多数的软件工程实践中，测试由于受限于时间成本和经济成本，是不可能去穷尽所有可能的组合的，而是采用苦于风险驱动的模式，有所侧重地选择测试范围和设计测试用例，以寻求缺陷风险和研发成本之间的平衡。

总结

首先，对于高质量的软件测试，用例设计不仅需要考虑明确的显式功能性需求，还要涉及兼容性、安全性和性能等一系列的非功能性需求，这些非功能性需求对软件系统的质量有着举足轻重的作用。

其次，优秀的测试工程师必须具有宽广的知识面，才能设计出有针对性、更易于发现问题的测试用例。

最后，软件溑匠用例设计是不可穷尽的，工程实践中难免受制于时间成本和经济成本，所以优秀的测试工程师要兼顾缺陷风险和研发成本之间的平衡。