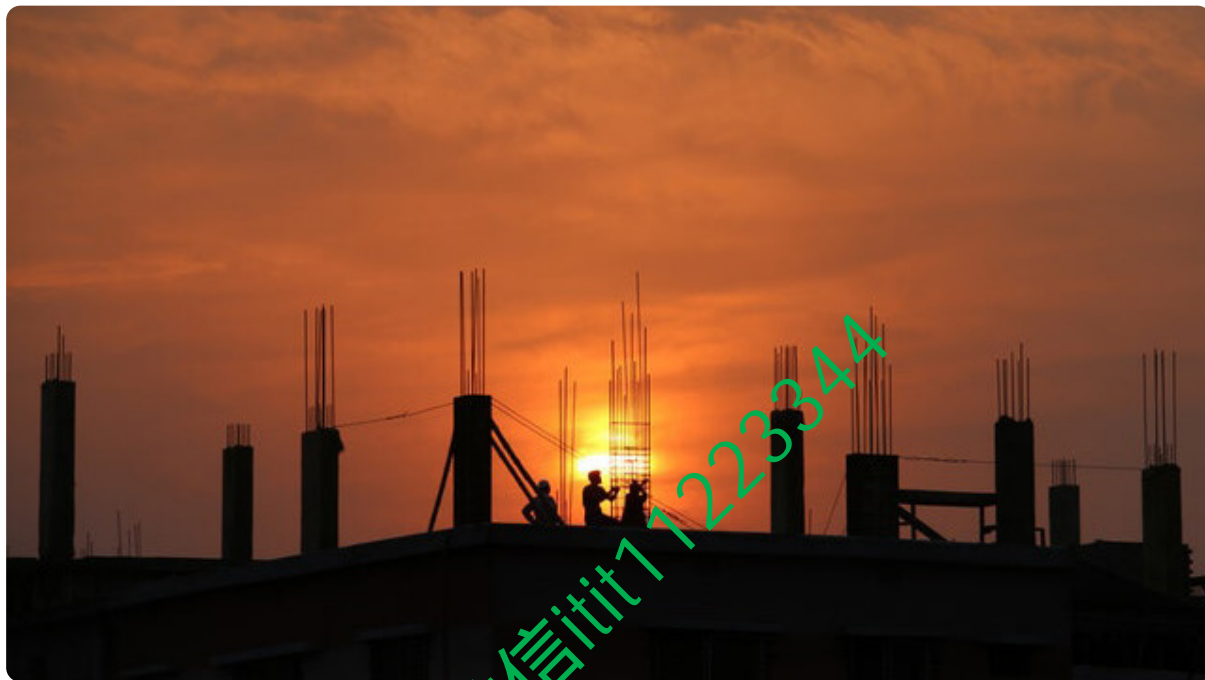


08 厉兵秣马：如何设计优秀的测试用例

更新时间：2019-09-10 10:35:00



“ 低头要有勇气，抬头要有底气。——韩寒 ”

作为软件测试里最基本最通用的技能，测试用例的设计相信大家都有概念，但是你真的能设计出优秀的测试用例么？我想应该打一个问号。我们把这个主题“庖丁解牛”一下，“设计优秀的测试用例”包含一个名词，一个动词还有一个形容词。最基本的首先是“测试用例”这个名词，我们从这里聊起。

测试用例

什么是测试用例？比较通用的解释：测试用例是通过使用在测试计划中确定的测试技术，对于已确定的测试条件进行逐步推敲，精炼而设计出来的重点说明如何具体操作产生何种结果的文档。翻译成通俗易懂的话就是你要把想要测试的动作变成在什么情况下，做什么动作，用什么数据方式去做，最后得到什么样的结果归集成一条条测试用例。所以，每个测试用例应该包含的要素各

举个例子，我们就拿最常用最通俗易懂的登录功能来说。基本上每个网站、每个 APP 都需要登录，那怎么成功登录呢？找一个已经存在的用户，在界面上输入正确的用户名和密码，点击一下登录按钮，看看有没有登录成功。OK，这是一个最最简单的操作，也构成了我们的第一个最正常的测试用例。仔细看看，其实也包含了前边说的几个要素：

- 前置条件：一个已经存在的用户
- 动作和参数：输入正确的用户名、密码，点击登录
- 期望结果：登录成功

这当然是一个完整的测试用例，但是对于一个登录模块来说，这就够了么？当然不，这只是这个模块里合格的测试用例之一。那该怎么写出更多的测试用例呢？这时候就要去看我们的动词了：设计。设计其实对应的是一套方法，只有有正确的方法，才能设计出合格的测试用例。

用例设计方法

工欲善其事必先利其器，做好一件事情的前提先找到有利的武器。磨刀不误砍柴工说的是同样的道理。在测试用例的设计上，比较常用的有下边几种武器。

- 等价类划分法
- 边界值法
- 因果图设计法
- 判定表设计法
- 正交实验法
- 错误推测法
- 场景法

具体每种方法到底是什么原理，原则是什么，怎样去设计，我相信聪明的你已经知道应该去 STFW 了。因为到处都讲的很多很清晰，但是几乎都是分开来说的，很多同学在学习了这些方法之后有一个疑问：测试的方法也太多了，我到底该怎么抉择呢？这里有一首非常好听的歌说明了这个问题：

输入分类选等价，
给定范围加边界，
条件组合出因果，
条件孤立想判定，
五阳六景取正六

错误推测靠经验，
测试充分全覆盖。

有没有跟着一起愉快的唱起来？好吧..... 其实我也没有，实在是找不到适合唱出来的曲调，但是实际上我们在真实设计用例的时候真的用得上这么多方法么？答案是否定的，在我相对还算漫长的测试生涯里，我觉得用的最多的就是**边界值**、**等价类**、**场景法**和**错误推测**这样几个方法。所以归纳总结一下，真正工作中我们设计的思路大概是：

- 用等价类划分方法划分大部分场景设计测试用例
- 任何情况下都必须使用边界值分析方法，经验表明用这种方法设计出测试用例发现程序错误的能力最强
- 程序业务复杂度比较高，则适当使用场景法补充一部分测试用例
- 如果你对该业务非常熟悉，可以根据经验在容易出错的地方补充一些测试用例

基于这样的设计方法，同样还是当初的那个登录界面，我们大体可以设计出下边的测试用例：

1. 已存在的用户，输入正确的用户名和密码，点击登录，验证是否登录成功；
2. 不输入任何内容，直接点击登录，验证是否登录失败，提示信息正确；
3. 已存在的用户，输入正确的用户名和错误的密码，点击登录，验证是否登录失败，提示信息正确；
4. 已存在的用户 A 与 B，输入 A 的用户名和 B 的密码，点击登录，验证是否登录失败，提示信息正确；输入不存在的用户名和任意密码，点击登录，验证是否登录失败，提示信息正确；
5. 已存在但状态异常的用户（如停用、冻结、锁定），输入正确的用户名和密码，点击登录，验证是否登录失败，提示信息正确；
6. 已存在的用户，用户名为小写，输入大写的用户名及正确的密码，点击登录，验证是否登录失败；
7. 已存在的用户，密码为小写，输入正确的用户名及大写的密码，点击登录，验证是否登录失败；
8. 密码是否自动加密显示或包含隐藏 / 显示功能，验证是否可以正常使用；
9. 用户权限是否区分，管理员及普通用户登录成功跳转是否正确；
10. 用户名及密码输入框是否具有长度限制，与注册时长度要求是否一致；
11. 登录失败到达一定次数，是否会自动显示验证码；
12. 有验证码情况下，输入正确的用户名密码及正确的验证码，验证是否登录成功；
13. 有验证码情况下，输入正确的用户名密码及错误的验证码，验证是否登录失败；

15. 有验证码情况下，点击验证码图片（或换一张按钮）更换验证码，使用更换前验证码登录，验证是否登录失败；
16. 刷新页面验证码是否跟随刷新；
17. 验证码超过可用时效，输入当前验证码，验证是否登录失败；
18. 登陆失败后，用户名是否保存，密码为空；
19. 是否有记住用户名功能和记住密码功能，是否可用；
20. 快捷键是否可用，密码是否不可以通过粘贴粘入；
21. TAB、ENTER 是否可以自动跳转控件及自动提交；
22. 是否支持第三方登录（微信、支付宝、QQ、微博等），登录验证是否正确；
23. 是否支持手机验证码登录，手机是否可以收到短信，是否可以登录成功；
24. 手机验证码超时，使用已超时验证码登录，是否可以登录成功；
25. 用户 session 失效后是否重新跳转登录页；
26. 用户登出后，通过后退按钮，是否可以继续操作；
27. 是否具有忘记密码功能，是否可用。

OK，简单列出了一些，但是如果这时候要我给这套用例打个分的话，那我可以打个 80 分。一定小伙伴要实名 diss 我了，已经这么全面了，还要什么？为什么只有 80 分？这就涉及我们今天讨论主题的第三个维度了，形容词“优秀的”。

谈优秀的测试用例

大家都清楚的一个道理，当你的分数是不及格的时候，要把自己提升到及格很容易，可以当你自己已经比较优秀了，想从 80 分提升到 90 分，那就很难了，这需要更深入的知识掌握。用例设计也是如此。同时，既然要谈“优秀”这个形容词，就要聊聊很多人的错误认知。很多人觉得对测试用例的评价在于表达清晰、条件 & 过程 & 预期覆盖完整、设计方法使用得当，这当然没有问题，但是当你用“优秀”的标准来评判的时候，就不要再看某一条用例了，而是要从整体上去考量。这就好像这两年火起来的一档节目《这就是街舞》，当你单人 battle 的时候，考量你的技术难度够不够高，够不够契合音乐，是不是炸，但是当齐舞比拼的时候，那考核的就是整体编舞、整齐度、故事性等等，对于一组测试用例来说，只有完备的、可重复的、可验证的、需求覆盖全面的测试用例才是最优秀的测试用例。

所以，作为老测试猿们，一定这时候明白了我要说什么。所谓的需求覆盖全面，所谓的完备一定不仅仅包含我们刚才的纯黑盒功能测试的内容，测试人员需要考虑的更多。一般情况下，我们还需要考虑数据日志测试、界面 UI 测试、兼容性测试、性能测试和安全测试等方面。我们分开来看：

所谓的数据日志测试主要包含我们在前端，在页面或者 APP 上看不到的测试项，我举几个例子：

- 数据库密码字段验证是否加密
- 登录失败次数是否记录在数据库、缓存中，逻辑是否正确
- 登录失败冻结等场景数据库是否正确修改状态
- 错误日志是否完备，是否便于排查问题
- 对象是否容易定位，便于自动化
- 是否有增加埋点，进行用户行为分析

界面 UI 测试

日常测试来说，界面和用户体验的测试也是非常必要的，像我们前边关于 TAB 和 ENTER 的使用其实就是用户体验的一种，所以如果要更全面地进行测试覆盖，就一定也要考虑到界面的测试。

- 布局是否合理，是否对齐
- 界面设计是否与需求、UI 设计文档一致
- 是否有错别字、标点错误缺失
- 页面颜色搭配是否得当
- 错误文字是否明确易懂
- 界面视觉效果是否恰当，界面动画展示是否流畅

兼容性测试

- 不同操作系统下，是否可以兼容（Windows、MAC OS、LINUX）
- 同一操作系统不同版本下，是否可以正常显示及功能正确性
- 不同浏览器下（Chrome、IE、FireFox 等）下是否可以兼容
- 同一浏览器不同版本下是否可以兼容
- 移动端是否兼容
- 放大缩小界面时是否兼容展示
- 不同语言下，界面展示是否正确
- 是否具有高对比读模式（为视力不好的人准备）

性能测试

性能测试又可以分为服务端性能和前端性能，也需要综合考虑，同时，针对性能的指标和场景也

- 用户登录接口的最大并发数（响应时间 3s 内）
- 特定负载测试下服务器的性能指标
- 压力测试过程中服务的稳定性和性能指标
- 服务的分布式处理逻辑，负载均衡逻辑、缓存及队列的使用
- 能够支持的接口最大并发量
- GC 处理，是否有内存溢出等情况
- 高并发下数据库是否有慢 SQL 和死锁
- 页面加载速度
- 页面资源大小，是否应用雪碧图
- YSLOW 分析，静态性能

安全测试

最后，也是我们最容易忽略的安全测试。很多时候我们容易忽略安全带来的威胁，但是实际上一旦发生安全问题，产生的损失会远远大于某一个 Bug 带来的影响。我们先就登录界面简单介绍一下应该采取的安全检测措施，后边有机会我们再细致聊一下安全测试的方方面面。

- 验证关键数据通过 HTTP 还是 HTTPS 传输
- 是否包含弱密码
- 是否容易被暴力破解
- 是否使用多因子认证
- 登录是否采用互斥性验证
- 产生的会话令牌（sessionID）强度
- 传输中是否存在会话令牌泄露情况
- 是否包含越权漏洞
- 使用万能密码是否可以登录成功
- 是否可以进行 SQL 盲注
- 密码存储加密安全性
- 是否具有 XSS（跨站脚本）攻击防御
- 是否包含 CSRF 攻击漏洞
- Token 或密码传输中防中间人攻击

到这里，我可以给上边的测试用例打上一个 95 分了，有 4 分是兼容可能存在的遗漏，最后 1 分是怕自己骄傲。可以看到，一个大家眼中很简单的登录功能，我们列出了如此多的测试用例，同时大家也可以看到，只有当你的“自我修养”足够强大，级别足够高，才能设计出更全面、更完

当然，我刚刚也说了，可能还会存在这些那些的遗漏，甚至有的同学会说：老板说测试时间紧，要很快上线，根本不可能执行这么多测试用例啊。这里我想说，没有人能够覆盖完整所有的场景，所以才会有如斯的用例设计方法；我们也没办法保证经过我们测试的项目、模块一定不出现任何问题，我们需要在**产品质量与人力成本**之间权衡，我们也需要在测试执行以前进行权衡，所以现在的测试用例增加了一个“**优先级**”的字段，便于我们更好的进行测试规划。

更可能有同学说：风落啊，你写这个很多我都不知道该怎么去测试啊，怎么办？不要着急，现阶段我们仅仅是抛砖引玉，继续看下去，希望在几个月之后，在这个专栏结束的时候，你可以找到更好的自己，也不要忘了前边介绍过的，“STFW”。最后，欢迎大家留言给风落，聊聊你在用例设计过程中有哪些“惊心动魄”的故事。

← 07 项目全流程中，tester们都在做什么？

09 论高质量的BUG →

精选留言 3

欢迎在这里发表留言，作者筛选后可公开显示

jooh123

老师，业务庞大的系统，刚入职不久业务不熟悉的情况下，设计用例考虑不到其他场景，现在只能依托需求表面进行用例设计，请问关于这个有啥好的建议或者改进吗

👍 1 回复

2020-03-03

风落几番 回复 jooh123

业务测试本身有一点，就是不熟悉的情况下可以开展测试，但不能负责设计负责把关，除非业务实在太简单。如果你去负责，那我觉得这是leader的失误。因为已经业务庞大的系统，必须有熟悉的人在负责把关，不然你想象一下余额宝团队突然所有老测试都撤退了，来个新人上手直接测试这是多么可怕的事儿。。你的问题不是技术问题，是流程问题

回复

2020-03-05 09:48:22

Yang11100

老师 庖丁解牛不是这个意思??

风落几番 回复 Yang11100

哇哈哈，果然有人看出来了~~~所以我之前偷偷加了个引号~ 找不出合适的成语来展现了~~

回复

2020-02-19 11:09:21

一席言

弱弱的问下，如果面试的时候提问给出一个界面或者一个功能，要设计测试用例像这种问题，如何设计测试用例具体从哪几个方面回答~从哪几个方面设计测试用例

👍 12

回复

2019-09-12

风落几番 回复 一席言

就像在专栏里说的。其实面试官考虑的不是你临时的设计是不是全，而是是不是有各方面的考虑，功能上从等价类到边界值到经验的错误分析，性能上、安全上、兼容性上、数据库上等等方面都要考虑到，并且表达出来，这样就好了。而不是直接说1、2、3、4.

回复

2019-09-16 10:48:52

NewBeQ 回复 风落几番

收获了，给老师和提这个问题的同学比心、点赞

回复

2020-02-06 14:51:10

手不如一看，干看不如一练

一学信信11223344