

软件测试用例

什么是测试用例

- 为特定测试目标或测试条件而制定的一组输入值、执行入口条件、预期结果和执行出口条件的集合

为什么需要测试用例

- 定义好测试活动，避免盲目测试、提高测试效率。

杀虫剂效应

随着被测软件的规模越来越大，功能越来越复杂，越来越多的缺陷开始出现，

测试工程师对其进行不断的进行测试、不断的回归，但仍然发现每次测试仍然会发现很多的缺陷（测试无穷尽）。

- 被测软件越来越大，功能越来越复杂（害虫抵抗力越来越强）
- 测试人员思维定势，使用测试技术和方法单一（长期使用同一款农药）

解决办法

- 交叉测试。测试团队成员对被测系统交叉模块测试。（使用不同品牌的农药）
- 测试人员提升自己能力，掌握新技能，新思想，新方案。用更多测试技术提高测试覆盖率。（修改农药配方，提升配方质量）
- 引入更高级测试人员，同时对现有技术人员进行技能培训。（提高使用农药浓度）

测试用例的意义

1.测试用例是软件测试的核心。

软件测试的重要性是毋庸置疑的，测试用例是测试工作的指导，是软件测试质量稳定的根本保障。

影响软件测试的因素很多，如软件本身的复杂程度，开发质量，测试方法和技术的运用。也有可观因素存在，如人员变动，环境，情绪等。

2.评估测试结果的基准

测试用例的通过率以及错误率，是测试结束的一个重要依据，用来判断该软件测试结果是否可以通过，能否达到上线的标准。

3.保证测试的时候不遗漏测试功能点。可以对测试工作进行牵引。

4.在编写测试用例的过程，可以熟悉需求，对系统架构或者业务流程有一个整体深入的了解。

设计测试用例

黑盒测试方法

- 等价类划分 (最常用)
- 边界值分析 (最常用)
- 因果图和决策表
- 状态转换图
- 基于经验的测试

白盒测试方法

- 语句覆盖
- 分支覆盖
- 条件覆盖
- 路径覆盖

测试用例的生命周期



测试环境设计

测试环境(TE)

为了运行被测软件、完成测试工作所必须的硬件、软件和网络环境的集合。稳定可控的测试环境可以使测试人员花费较少的时间，完成测试用例的执行。

测试环境内容

硬件环境

软件环境

网络环境

测试数据

测试工具

测试环境设计原则

- 尽可能用真实的环境，少用模拟器和虚拟机。
- 机器配置根据软件不同，不得低于软件运行最低要求。
- 选择主流的操作系统以及软件平台，例如ios系统 Android系统。
- 测试环境要干净、独立、排除干扰。(例如无用的杀毒软件，无用的播放器等等，性能测试中，忽然蹦出个洞修复程序，那必然影响测试结果)

测试环境所需知识技能

- 常见操作系统安装使用
- 安装程序所需驱动，解决驱动报错问题
- 数据库使用
- 浏览器调试
- 模拟器和虚拟机的使用
- 系统镜像和备份与还原工具的使用

软件测试工具

最常见的如下:

测试管理工具(项目流程管理)

- 惠普HP QC TD
- 国外工具 JIRA 使用最多，易用性
- 国内工具 禅道

功能测试工具(自动化脚本测试)(黑盒)

- 惠普HP QTP(脚本录制工具，解放重复的点击等工作)， vbs语言脚本
- Selenium

性能测试工具(黑盒)

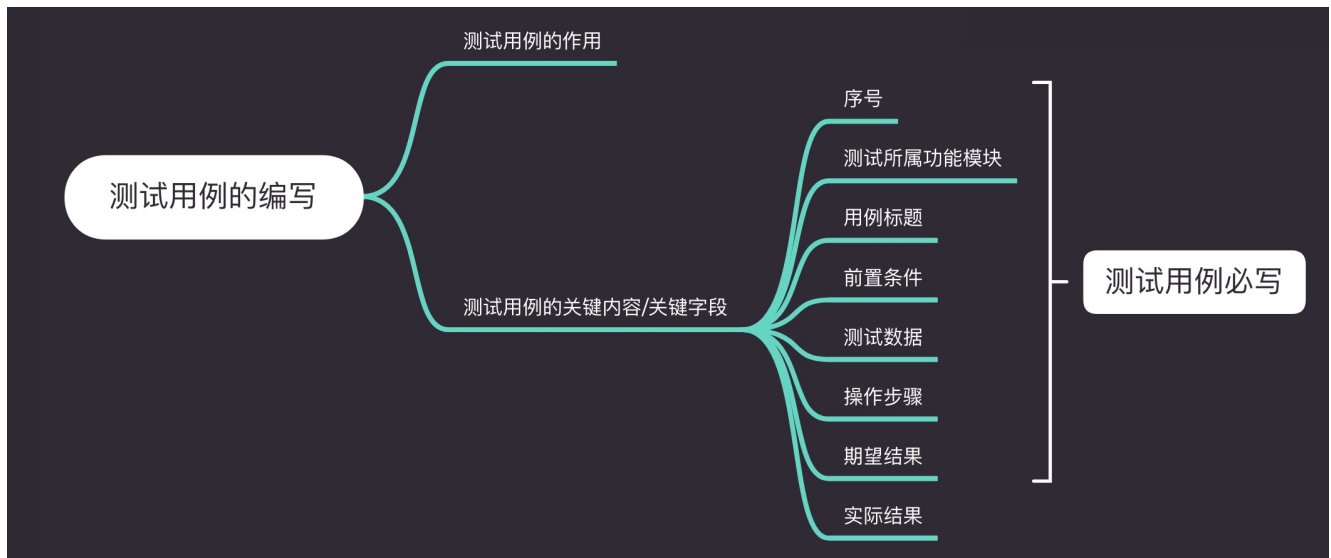
- HP LoadRunner (使用人数最多，易用，收费软件)
- Apache Jmeter(免费)

代码测试工具(白盒测试)

- JUnit

测试用例的八大要素

- 1.用例编号：产品名字-测试阶段
- 2.测试项目：对应一个功能模块
- 3.测试标题：直接对测试点进行细化得出
- 4.重要级别：高/中/低
- 5.预置条件：需要满足一些前提条件，否则用例无法执行
- 6.测试输入：需要加工的输入信息，根据实际情况设计
- 7.操作步骤：明确给出每个步骤的描述，执行人员根据该步骤执行工作
- 8.预期结果：根据预期输出对比实际结果，判断被测对象是否符合需求。
- 9.实际结果：根据实际结果，填写报告。(可写可不写)



测试用例文档

	A	B	C	D	E	F	G	H	I	J	K
1	序号	模块名称	测试子项	测试意图（用例名称）	用例级别	预置条件	测试步骤	预期结果	测试结果	缺陷编号	备注信息
2											

黑盒测试方法

黑盒测试：是一种常用的软件测试方法，它将被测软件看作一个打不开的黑盒，主要根据功能需求设计测试用例，进行测试。

几种常用的黑盒测试方法有 等价类划分法、边界值分析法、因果图法、决策表法。在实际运用中要选择合适的方

等价类划分法是一种典型的黑盒测试用例设计方法。采用等价类划分法时，完全不用考虑程序内部结构，设

计测试用例的唯一依据是软件需求规格说明书。

等价类划分法：

- 等价类中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。
- 后续我们只要从每个等价类中任意选取一个值进行测试，就可以用少量具有代表性的测试输入取得较好的测试覆盖结果。

等价类划分 有效等价 和 无效等价 。

等价类：某个输入的集合，集合中每个输入条件都是等效的，如果其中一个输入无法发现问题，集合中其它输入也是无效的

有效等价类：合理的输入数据

无效等价类：不合理的输入数据

所谓等价类，是输入条件的一个子集合，该输入集合中的数据对于揭示程序中的错误是等价的。等价类又分为有效等价类和无效等价类。有效等价类代表对程序有效的输入，而无效等价类则是其他任何可能的输入（即不正确的输入值）。

有效等价类和无效等价类都是使用等价类划分法设计用例时所必须的，因为被测程序若是正确的，就应该既能接受有效的输入，也能接受无效输入的考验。

等价类划分的原则

- 如果输入条件规定了一个取值范围，那么就应该确定一个有效等价类以及两个无效等价类。
- 规定了输入值的集合必须如何的情况下可以确定一个有效等价类和一个无效等价类。
- 在输入数据是一个布尔值的情况下，可以确定一个有效等价类和一个无效等价类。
- 已经划分的等价类中各个元素在程序中的处理方式不同，可以将等价类再次细分
- 在规定了输入条件必须遵守一定规则轻客下，可以确定一个等价类（符合规则）与多个无效等价类

等价类划分法设计测试用例步骤

- 为每个输入划分等价类，且标记唯一编号
- 设计测试用例，使其尽可能多的覆盖尚未覆盖的有效等价类，重复步骤，使得所有有效等价类被覆盖。
- 设计测试用例，使其只覆盖一个无效等价类，重复步骤，使得所有无效等价类被覆盖。

驾考年龄限制



驾驶证年龄新规

对于小型手动(C1驾驶证)、自动(C2驾驶证)、残疾人专用自动挡载客汽车(C5驾驶证)的，驾考年龄为18—70周岁。

1. 有效等价类：驾驶人年纪18~70整数
- 2.
3. 无效等价类：年纪<18；年纪>70

网易邮箱注册

欢迎注册网易邮箱！邮件地址可以登录使用其他网易旗下产品。

注册免费邮箱

注册VIP邮箱

* 邮件地址

@ 163.com ▾

6~18个字符，可使用字母、数字、下划线，需以字母开头

* 密码

6~16个字符，区分大小写

* 确认密码

请再次填写密码

* 验证码



请填写图片中的字符，不区分大小写

[看不清楚？换张图片](#)

* 手机号码

 +86-

请编辑短信：222发送到10690163222，以确保账号安全。
(短信费用由运营商收取)

☐ 同意《网易邮箱帐号服务条款》和《网易隐私政策》

已发送短信验证，立即注册

账号限制条件

6~18个字符，可以使用字母、数字、下划线、需以字母开头

输入条件		有效等价类	无效等价类	
邮箱地址@163.com		6<字符数<18 ¹	字符数>18 ² ；字符数<6 ³ ；零个字符 ⁴	
		使用字母、数字、下划线 ⁵	特殊符号 ⁶ 、中文 ⁷ 、空格 ⁸	
		且以字母开头 ⁹	非字母开头 ¹⁰	
测试用例				
编号	输入数据	预期结果	覆盖等价类	
	1 alex	错误	3, 5, 9	
	2 alex_666	正确	1, 5, 9	
	3 alex_6666666666666666	错误	2, 5, 9	
	4 NULL	错误	4	
	5 alex^%\$12323	错误	1, 6, 9, 5	
	6 alex 666	错误	1, 5	
	7 666alex	错误	1, 5, 10	

实战案例一

微信红包，可以发送的金额范围

- 有效金额范围：0.01-200
- 有效类型：数字有效范围0.01<有效值 < 200
- 无效类型：字母，特殊符号，中文

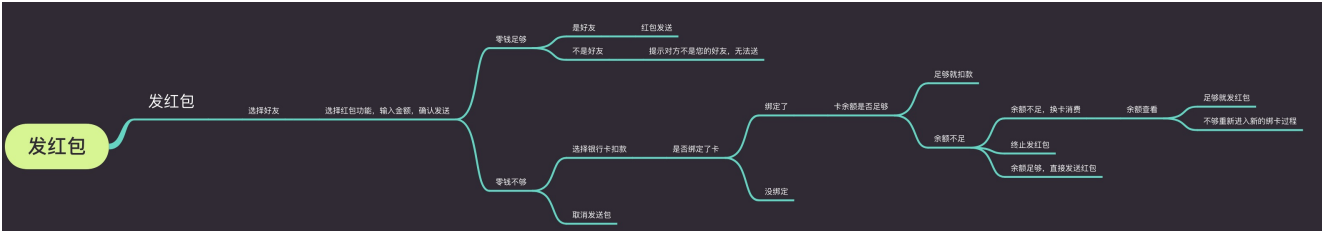
序号	功能项	有效等价类	编号	无效等价类
1	微信发红包	数字0.01<=钱包余额<=200	2	字母
				特殊符号
				中文

边界值分析法

边界值分析法是一种补充等价划分的测试用例设计法则，它不是选择等价类的任意元素，而是选择等价类边界的测试用例。

- 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。



实战案例二

学生管理系统中，可以录入考试成绩，成绩范围在0~100，考试成绩合格分是60分。

为了测试这个输入框，当然不会从0~100挨个测试，根据需求可知，0~59任意整数，60~100任意整数可以看做是等价的，都可以揭露输入框潜在的缺陷。

等价类划分法

在0~59之间和60~100之间随机抽取整数验证，就成为“有效等价类”。

显然如果输入的成绩是负数，或是大于100的数，都成为“无效等价类”。

因此，测试用例的设计：

有效等价类

- 0~59之间的任意整数
- 59~100任意整数

无效等价类

- 小于0的负数
- 大于100的整数
- 0~100之间的任意浮点数
- 其他任意非数字字符

序号	功能项	有效等价类	编号	无效等价类
1	学生成绩录入	整数0~59	2	小于0的数字
		59~100整数		大于100的数字
				0~100之间任意浮点数
				其他任意非整数数字字符

边界值分析法

我们继续看学生信息系统中“考试成绩”的例子，选取的边界值数据应该包括：-1，0，1，59，60，61，99，100，101。

序号	功能项	输入数值	被测边界	预期输出
1	学生考试成绩	-1 0 1	0	
		59 60 61	60	
		99 100 101	100	

因果图

等价类划分法和边界值分析方法都是着重考虑输入条件，而不考虑输入条件的各种组合、输入条件之间的相互约束关系。

如果输入之间有关系，例如，约束关系、组合关系，这种关系用等价类划分和边界值分析是很难描述的，测试效果难以保障，因此必须考虑使用一种适合于描述对于多种条件的组合，产生多个相应动作的测试方法，因果图正是在此背景下提出的。

比如注册功能，填写地区信息，条件一输入了北京地区，条件二的输入只能是昌平、海淀、朝阳等信息，这就是种约束关系。

因果图(逻辑模型)是一种适合描述多种条件的组合、产生多种相应动作的测试方法，这就需要用因果图。

因果图-判定表

- 因果图基于这样的思想：一些程序的功能可以用 **决策表** 的形式来表示，并根据输入条件的组合规定相应的操作。
- 为 **决策表** 中的每一列设计一个测试用例，以便测试程序在输入条件的某种组合下的输出是否正确。
- 因果图就是从程序规格说明书的描述中找出因(输入条件)-果(输出结果或程序状态的改变)
- 判定表可以理解是个表格，是分析和表达多逻辑条件下执行不同操作的工具
- 判定表可以把复杂的逻辑关系和多种条件组合的情况表达的非常具体

		1	2	3	4	5	6	7	8		
问题	你累了吗?	yes	yes	yes	yes	no	no	no	no		
	你对书籍内容感兴趣吗?	yes	yes	no	no	yes	yes	no	no		
	书籍内容会使你糊涂吗?	yes	no	yes	no	yes	no	yes	no		
建议	请回头重新阅读	√				√					
	继续阅读		√				√				
	跳过本章节								√		
	停止阅读，洗洗睡吧			√	√			√			

判定表通常由四个部分

- 条件桩：列出问题的所有条件(如上表，你累了吗)
 - 条件项：列出针对条件的取值，所有可能情况的真假值
- 动作桩：已列出的问题可采取的操作(停止阅读，洗洗睡吧)
 - 在条件项的情况下应采取的动作

因果图案例实战

两位数加法计算器

输入第一个数值

输入第二个数值

总和

+

分析输入条件与输出条件

输入1与输入2的数值区间：

- $0 \leq x \leq 99$
- $-99 \leq x < 0$
- $x < -99$
- $x > 99$

输出信息

- 正确计算结果
- 错误输入

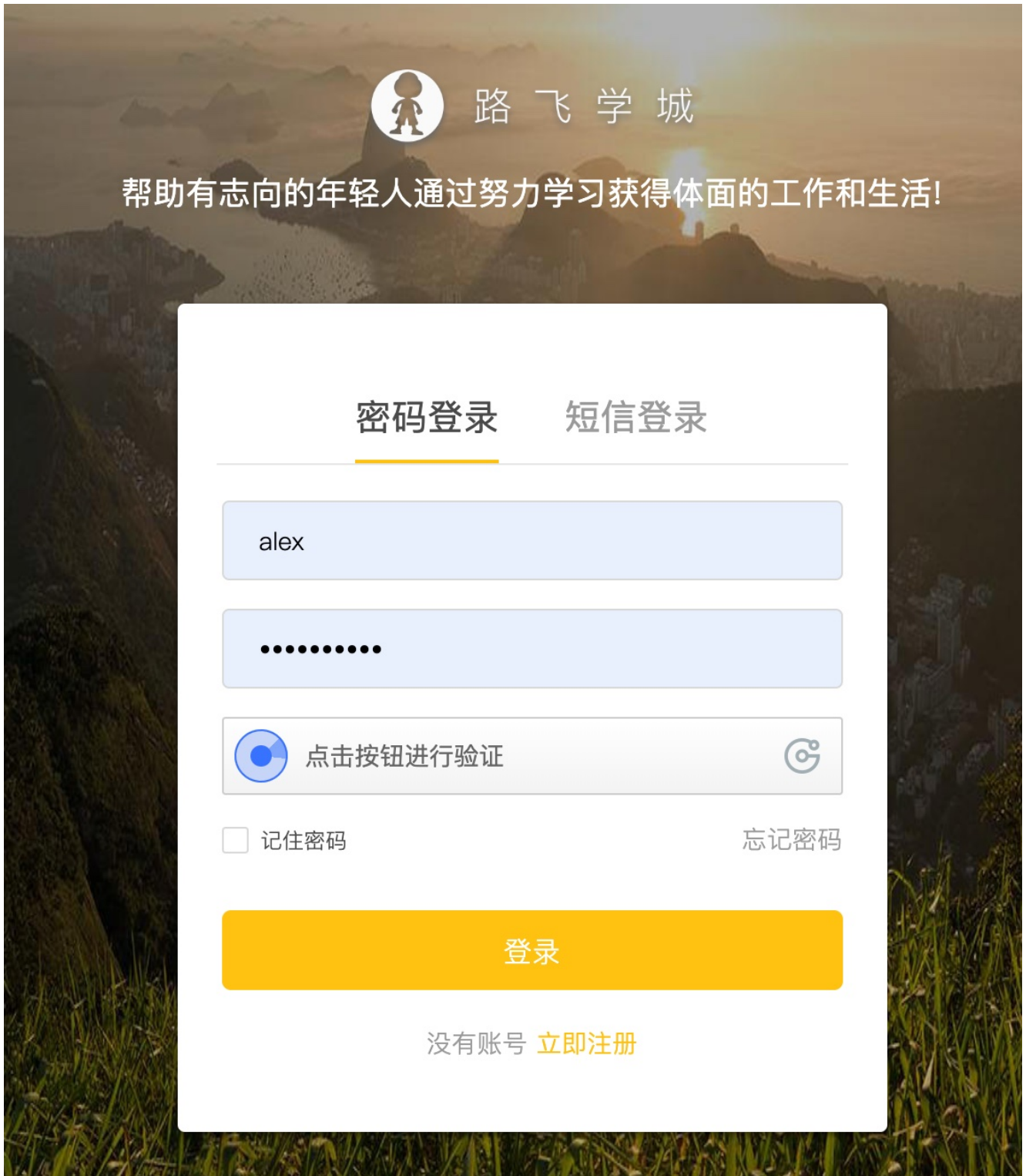
判定表绘制

输入数值1								
条件1	T	T	-	-	-	-	-	-
条件2	-	-	T	T	-	-	-	-
条件3	-	-	-	-	T	-	-	-
条件4	-	-	-	-	-	T	-	-
输入数值2								
条件1	T	-	T	-	-	-	-	-
条件2	-	T	-	T	-	-	-	-
条件3	-	-	-	-	-	-	T	-
条件4	-	-	-	-	-	-	-	T
输出结果								
正确计算	√	√	√	√				
错误提示					√	√	√	√

得到测试用例

输入数值1								
条件1	T	T	-	-	-	-	-	-
条件2	-	-	T	T	-	-	-	-
条件3	-	-	-	-	T	-	-	-
条件4	-	-	-	-	-	T	-	-
输入数值2								
条件1	T	-	T	-	-	-	-	-
条件2	-	T	-	T	-	-	-	-
条件3	-	-	-	-	-	-	T	-
条件4	-	-	-	-	-	-	-	T
输出结果								
正确计算	√	√	√	√				
错误提示					√	√	√	√

从“用户登录”谈起





路飞学城

帮助有志向的年轻人通过努力学习获得体面的工作和生活!

密码登录 短信登录

alex

.....

 点击按钮进行验证 

☐ 记住密码 忘记密码

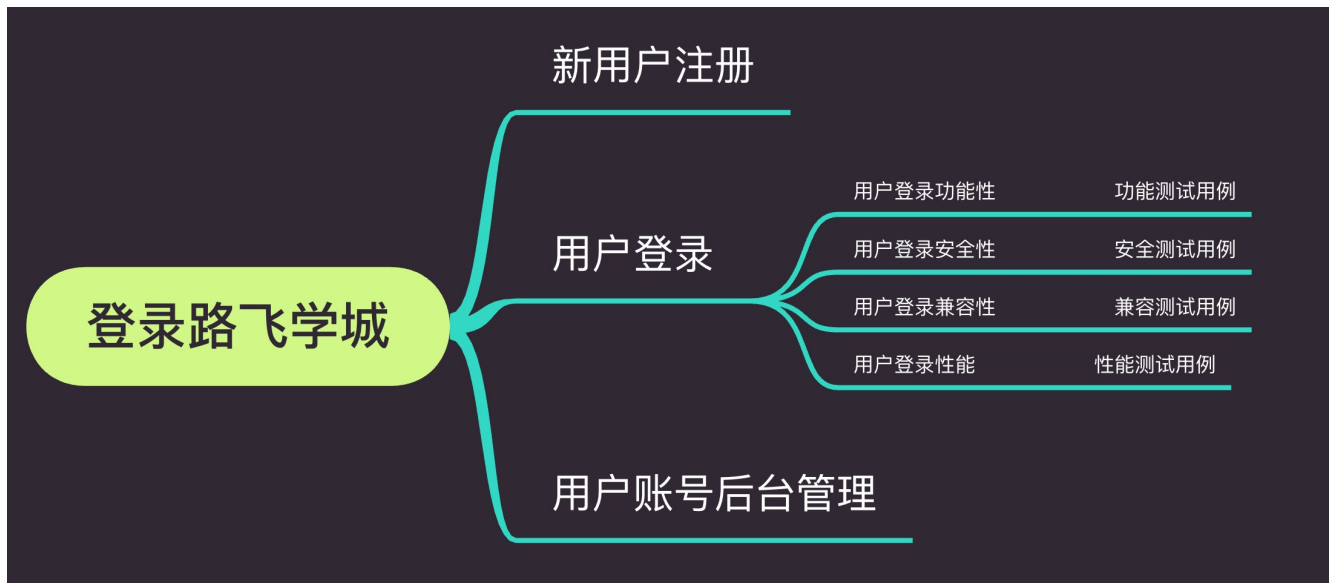
登录

没有账号 [立即注册](#)

用户登录，是我们耳熟能详的基本功能，基本上你接触到的网站都需要登录，我们只需要输入用户名和密码，然后点击登录按钮，查看页面是否登录。这就是最基本，最典型的一种测试用例。

作为测试工程师，**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。**

在具体的用例设计时，首先需要搞清楚每一个 **业务需求** 所对应的多个 **软件功能需求点**，然后分析出每个软件功能需求点对应的多个 **测试需求点**，最后再针对每个 **测试需求点** 设计 **测试用例**。



有趣的面试题

面试官可能会问：“一个登陆页面，你给我设计一些测试用例吧。”

新人测试工程师的思路

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

- 输入已注册的用户名和正确的密码，验证是否登录成功；
- 1. 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；
- 2. 输入未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；
- 3. 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；
- 4. 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；
- 5. 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；
- 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，输入错误的验证码，验证是否登录失败，并且提示信息正确。

经验丰富测试工程师的思路

- 用户名和密码是否大小写敏感；
- 页面上的密码框是否加密显示；
- 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；
- 忘记用户名和忘记密码的功能是否可用；
- 前端页面是否根据设计要求限制用户名和密码长度；
- 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；
- 刷新页面是否会刷新验证码；
- 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；
- 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；
- 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；
- 页面默认焦点是否定位在用户名的输入框中；

- 快捷键Tab和Enter等，是否可以正常使用。

上面所有的测试用例设计都是围绕显式功能性需求的验证展开的，换句话说，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的。

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)

- 指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，这都是属于典型的显式功能性需求描述。

非功能性需求 (Non-functional requirement)

- 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。

安全性测试用例

- 用户密码后台存储是否加密；
- 用户密码在网络传输过程中是否加密；
- 密码是否具有有效期，密码有效期到期后，是否提示需要修改密码；
- 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；
- 密码输入框是否不支持复制和粘贴；
- 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；
- 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；
- 用户名和密码的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；
- 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；
- 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；
- 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

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

兼容性测试用例

- 不同浏览器下，验证登录页面的显示以及功能正确性；
- 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；
- 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；
- 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。

可见，一个普通的登录功能，在不同的考虑场景下，居然涵盖如此多的测试用例，一个优秀的测试工程师必须具有很宽广的知识面，能够对被测系统的深入理解，明白安全攻击基本原理，掌握性能测试基本方法，才

能设计出有的放矢的测试用例

穷尽测试

尽管我们提出了如此多的测试用例，一定还会有未涉及到的所有测试方法。

穷尽测试指的是包含了软件输入值和前提条件所有的可能的组合的测试方法，完成穷尽测试的系统中不该残留任何未知的软件缺陷，因为必须去做更多的测试去找到他们。

但是在绝大多数软件中，测试受到时间和经济成本的影响，不可能完成穷尽测试，而是基于风险驱动模式，侧重的设计测试用例，寻求缺陷和研发成本的平衡。

经典面试题之水杯测试用例

在软件测试的面试中，经常会碰到类似的问题，比如： 如何测试一个杯子，或者如何 测试一只笔。

要求你设计20个以上的测试用例。

这类的面试题目，是考察面试者是否熟悉各种软件测试方法，设计test case的能力。

回答这类问题的思路，应该从软件测试的各种不同方法来联想，具体如下

- 功能测试角度
 - 能否盛水
 - 能否装其他液体，如雪碧，醋，硫酸，茶叶
 - 杯子的容量
 - 杯子的材质，纸质，玻璃，塑料
 - 杯子是否有异味
 -
- 界面测试
 - 杯子客户是小孩，还是老人，还是年轻人，杯子封面设计
 - 杯子颜色
 - 杯子形状
 -
- 安全性测试
 - 杯子材料是否通过了可食用
 - 是否结实，摔在地板上，水泥地，土地，是否会坏
 - 是否容易破损，缺口是否会伤人
 - 是否能够微波
 -
- 性能测试
 - 是否耐热
 - 是否耐寒
 - 如果纸质的，盛水几天后会漏水
 - 杯子上的图案多久会掉色
 - ...
- 易用性测试
 - 杯子是否烫手
 - 杯子把手是否好拿

- 杯子是否防滑
- 杯子是否容易喝到

美团测试用例模板

<https://tech.meituan.com/2016/03/22/testcase-template.html>

(<https://tech.meituan.com/2016/03/22/testcase-template.html>)

🗨 用户评论

▲ 登录 (/login/) 或 注册 (/register/) 后才能发表评论

目录

- 预科：为什么学测试 (/wiki/detail/8/68/)
- 什么是计算机 (/wiki/detail/8/69/)
- 第一章 测试理论 (/wiki/detail/8/70/)
- 软件测试方法 (/wiki/detail/8/71/)
- 软件测试需求 (/wiki/detail/8/72/)
- 软件测试计划书 (/wiki/detail/8/73/)
- **测试用例□ (/wiki/detail/8/74/)**
 - 软件测试用例
 - 什么是测试用例
 - 为什么需要测试用例
 - 测试用例的意义
 - 设计测试用例
 - 测试用例的生命周期
 - 测试环境设计
 - 测试环境内容
 - 测试环境设计原则
 - 测试环境所需知识技能
 - 软件测试工具
 - 测试用例的八大要素
 - 黑盒测试方法
 - **等价类划分法：**
 - 等价类划分的原则
 - 等价类划分法设计测试用例步骤
 - 实战案例一
 - 边界值分析法

◁ 实战案例一

▼ 大纲索引

- 等价类划分法
- 边界值分析法
- 因果图
- 从“用户登录”谈起
- 有趣的面试题
 - **显式功能性需求 (Functional requirement)**
 - 非功能性需求 (Non-functional requirement)
 - 穷尽测试
- 经典面试题之水杯测试用例
- 美团测试用例模板
- 缺陷管理 (/wiki/detail/8/75/)
- 配置管理 (/wiki/detail/8/76/)
- 禅道 (/wiki/detail/8/77/)
- 软件质量 (/wiki/detail/8/78/)