

软件测试教程 第四节 用例篇

上一节课讲述的是一次基本的测试过程。在我们开始做了一段时间基础测试，熟悉了业务之后，往往会分配来写测试用例，并且在日常测试中，有时也需要补充测试用例到现有的案例库中。

在这里我们将回答以下问题

- 测试用例的基本要素
- 测试用例的设计方法
 - 基于需求的设计方法
 - 等价类
 - 边界值
 - 因果图
 - 正交排列
 - 场景设计法
 - 错误猜测法
- 测试用例的有效性
- 测试用例的粒度和评价

测试用例的基本要素

回顾测试用例的概念：

测试用例（Test Case）是为了实施测试而向被测试的系统提供的一组集合，这组集合包含：测试环境、操作步骤、测试数据、预期结果等要素。

好的测试用例是一个不熟悉业务的人也能依据用例来很快的进行测试

评价测试用例的标准：对比好坏用例的评价标准

- 用例表达清楚，无二义性。。
- 用例可操作性强。
- 用例的输入与输出明确。一条用例只有一个预期结果。
- 用例的可维护性好。
- 用例对需求的覆盖率高。

测试用例的给我们带来的好处

测试执行者的依据

使得工作可重复，自动化测试的基础

评估需求覆盖率

用例的复用

积累测试的方法思路以供后续借鉴

- 使用中带来困扰

测试用例的设计是费时费力的工作，往往设计测试用例所花费的时间比执行所花费的时间还多。

- 解决如下问题：

不知道是否较全面的测试了所有功能 测试的覆盖率无法衡量 对新版本的重复测试很难实施 存在大量冗余测试影响测试效率

测试用例的设计方法

基于需求进行测试用例的设计

基于需求设计测试用例是测试设计和开发测试用例的基础，第一步就要分析测试需求，验证需求是否正确、完整、无二义性，并且逻辑自治。在需求正确的基础上细化测试需求，从测试需求提炼出一个个测试点或者测试项，然后根据每一个测试点进行测试用例的设计；

在分析测试需求时，一般分为**功能测试需求**和**非功能测试需求**

功能需求测试分析

对于功能测试中，可以借助功能框图来帮助我们进行测试的需求分析。概括起来，功能测试需求包括以下，通常包括以下几个方面。

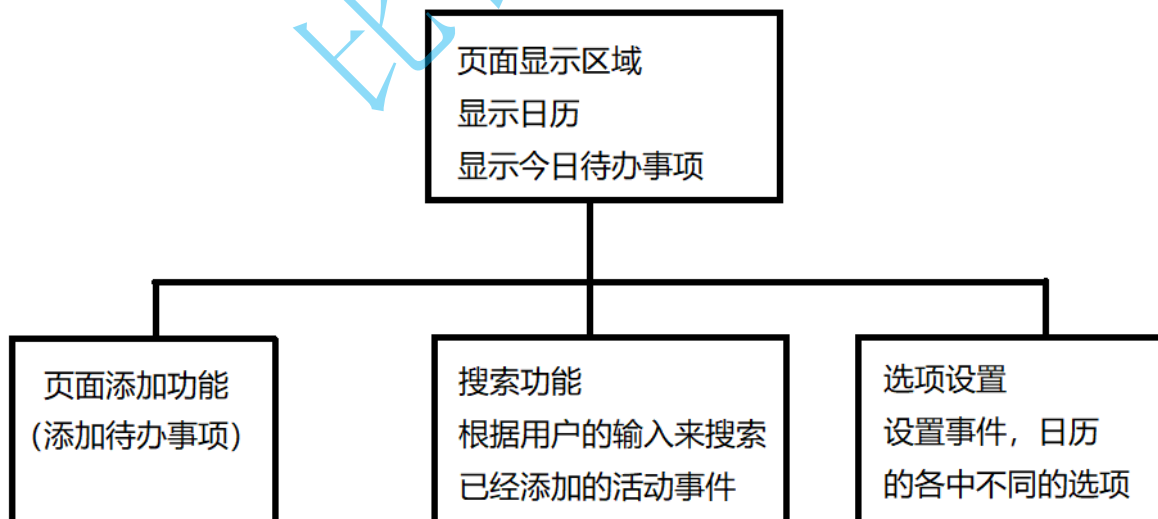
- (1) 系统各个功能界面的验证
- (2) 借助业务把功能串起来进行测试
- (3) 功能的一致性，交互性（多功能互操作）的测试
- (4) 系统的不同输入，结果输出的业务数据测试。
- (5) 功能的错误操作，异常操作的测试（属于负面测试）
- (6) 功能实现用到的算法验证，有时需要用源代码评审
- (7) 用户操作的易用性，用户体验，往往结合功能测试同时验证

针对具体的需求，可以根据业务分类，用户角色（餐厅的会员系统）或者用户操作区域等将系统的功能分解成若干个功能模块，然后按照功能模块分别进行测试需求分析。按照功能模块划分，业务模块划分是最常见的做法。

下面对一个简单的日历系统的需求进行分析

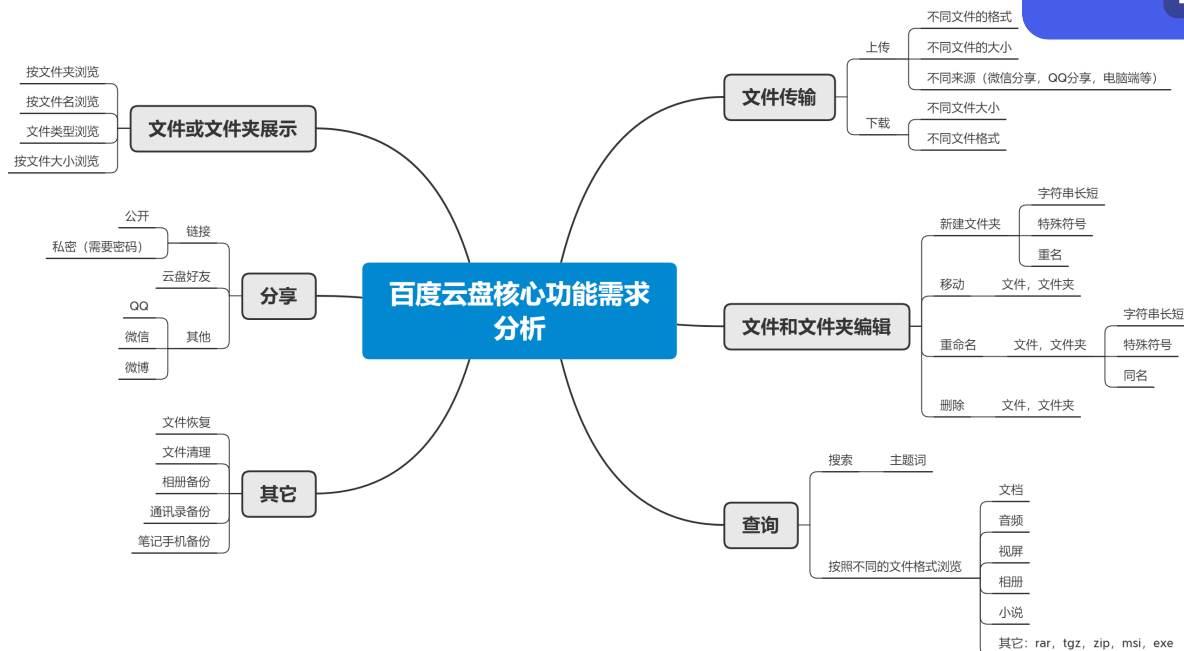


对日历根据web界面的功能布局分析出的功能框图如下：



也可以采用思维导图的方式，更为方便，有效，只管呈现测试需求的分析结果，可以更好的支持测试分析思路的连贯性。

下面以我们常用的**百度云盘**手机端为例进行分析功能



在进行需求分析的时候，我们还要考虑业务规则如，上传文件的大小有没有限制；一次性崔铎上传多少数量的文件，比如小于100个；文件夹最多有多少层等等；

非功能需求测试分析

非功能测试需求主要涉及性能，安全性，可靠性，兼容性，易维护性和可移植性等。从测试需求分析来看，每一类非功能特性测试都需要根据需求单独分析。他们之间可能会存在相互影响，如安全性越高，就越有可能给易用性，性能带来更大的挑战。

这里要说明的是对于每一个应用软件系统，非功能特性的质量需求都是存在的，但是不同的项目类型对各个非功能特性的要求是不一样的，这个需要根据具体的项目、具体需求和不同产品应用的特点进行分析。

(1) 纯客户端软件，比如字处理软件（Word，PPT），媒体（音频/视频）播放软件（电脑自带的）等。这类软件对系统的功能测试要求是最低的，但是对兼容性和稳定性，可移植性有一定的要求。

(2) 企业内部的客户端/服务端（C/S）应用系统。比如电子邮件，即时通信系统（飞Q，企业QQ）等，在系统功能测试需求上比纯客户端复杂，要求功能正确，稳定性能好。但是整体上看，对性能，安全性，兼容性要求不高。

(3) 外部大型复杂网络应用系统，比如电子商务，网上银行，视频网站（腾讯，优酷）等，除了有复杂的系统的功能测试需求外，在系统的性能，安全性，兼容性，容错性，可靠性等都有很高的要求。

此外，对于大型企业级应用系统，由于应用模式，系统架构的不同（分布式，微服务等），我们必须结合架构和应用模式来具体分析非功能性测试需求，特别是可扩展性，可靠性，安全性等。技术架构对功能的影响小，但是非功能性测试就要深入架构分析，才能更好的把我测试范围和测试方法。

给同学们继续举一个例子百度云盘非功能测试的

案例：

用户需求：

购买3000块钱以内的华为智能手机

测试用例：

1. 价格 \leq 3000元
2. 品牌为华为
3. 智能手机
4. 手机功能验证：
 - 4-1. 打电话
 - 4-2. 接电话
 - 4-3. 发短信
 - 4-4. 收短信
 - ...

软件需求：

1.1.1.1.5.3 异常事件流

1. 若用户未收到激活邮件，可在登录界面录入电子邮件及密码后，再次发送激活邮件。
2. 每次发送的激活邮件，仅在发送邮件后起24小时之内有效，超过24小时后需重新发送激活邮件。

测试用例

- 1-1、未收到邮件，登录时输入电子邮件及密码后，再次发送激活邮件
- 1-2、已收到邮件，登录时输入电子邮件及密码后，不发送激活邮件

- 2-1、收到邮件，未激活，24小时内进行激活
- 2-2、收到邮件，未激活，24小时后链接过期进行激活。
- 2-3、收到邮件，已激活，24小时后链接过期，再次点击激活？

页面检查：

- 1、收到激活邮件
- 2、邮件内容正确
- 3、激活URL正确，可激活
- 4、再次激活提示已激活
- 5、过期激活提示已过期

具体的设计方法

等价类

因材施教的例子：

原则上讲，老师应该依据每个学生自身的情况，指定符合的学习方案。但是实际上学生太多老师管不过来，只能分成几类：优等生强调知识面的扩展和综合能力的提升；中等生强调夯实基础，查缺补漏；差等生强调优先掌握重点，暂时跳过难点...

思路：输入的集合是无穷的，不能全都覆盖到

依据需求将输入（特殊情况下会考虑输出）划分为若干个等价类，从等价类中选出一个测试用例，如果这个测试用例测试通过，则认为所代表的等价类测试通过，这样就可以用较少的测试用例达到尽量多的功能覆盖，解决了不能穷举测试的问题。

- 有效等价类：对于程序的规格说明书是合理的、有意义的输入数据构成的集合，利用有效等价类验证程序是否实现了规格说明中所规定的功能和性能
- 无效等价类：根据需求说明书，不满足需求的集合。

等价类只考虑输入域的分类，没有考虑输入域的组合，需要其他的设计方法和补充。

超市买水果

有效等价类：苹果、桃子、梨

无效等价类：青菜、米、饮料，...

以上述的需求为例：

| 用户名 | 必填，录入用户名 | 6至15 | 字符类型A-Z,不区分大小写 |

用户名由长度为6-15位的字符串组成，那么针对字符有效等价类为A-Z,a-z，无效等价类为两个：数字：1, 0.1, -1 特殊字符：@, #, ¥, 空格

思考一下：如何针对6-15位长度设计测试用例？

边界值

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

日常语言中的"边界"漏洞

考完试发成绩了，老师布置寒假作业：超过60分的，所有题目抄写1遍，低于60分的，所有题目抄写3遍。

于是小明就没有写作业~~，因为他刚好60分。

1. 输入框长度为1-11，取边界值为：1、11、12、0
2. 运动员的参赛项目为1-3项，取边界值为：0项、1项、3项、4项
3. 查询面页面有999行，每50行为一页，取边界值为：输出0行、1行、50行、51行、999行

以注册邮箱的软件需求为例子

用户名要求长度为6-15位

边界值上点为：5, 6, 15, 16 全了吗？

在实际的测试设计中，会将等价类和边界值结合起来使用，那么我们最终可以确认的用例设计为：

5,6,10,15,16五个长度的字符的输入值

继续思考：这样的测试真的完整了么？中文？半角？全角？特殊字符(扩散思维)

错误猜测法

错误猜测法是对被测试软件设计的理解，过往经验以及个人直觉，推测出软件可能存在的缺陷，从而针对性地设计测试用例的方法。

这个方法强调的是对被测试软件的需求理解以及设计实现的细节把握，还有个人的经验和直觉。

错误推测法和目前流行的“探索式测试方法”的基本思想一致，这类方法在敏捷开发模式下的投入产出比很高，被广泛应运于测试。

这个方法的缺点是难以系统化，并且过度依赖个人能力。

案例：



以注册为例

- 1、校验中特殊字符空格的处理？
- 2、密码校验中的大小写？
- 3、姓名中的特殊字符？
- 4、密码发送是否明文

场景设计法

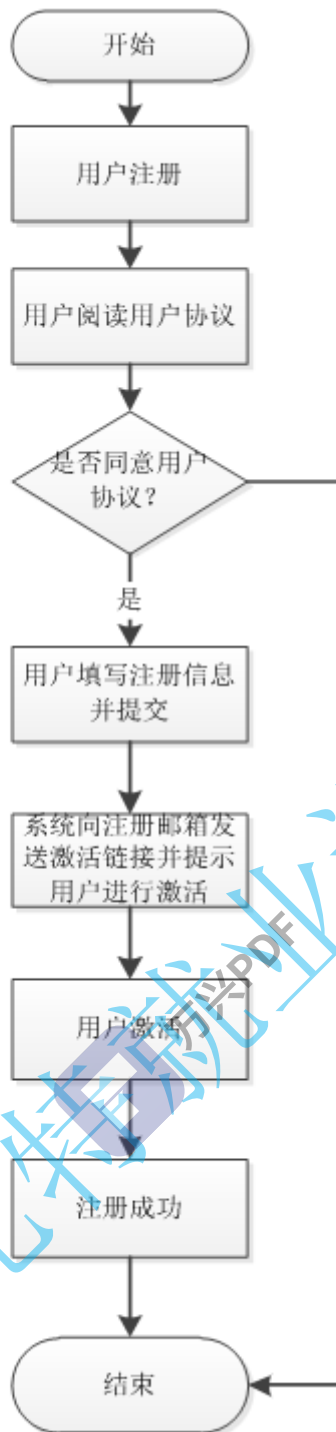
现在的软件几乎都是用事件触发来控制流程的，事件触发时的情景便形成了场景，而同一事件不同的触发顺序和处理结果就形成事件流。该方法可以比较生动地描绘出事件触发时的情景，有利于测试设计者设计测试用例，是测试用例更容易理解和执行。

典型的应用是用业务流把各个孤立的功能点串起来，为测试人员建立整体业务感觉，从而避免陷入功能细节忽视业务流程要点的错误倾向

案例：

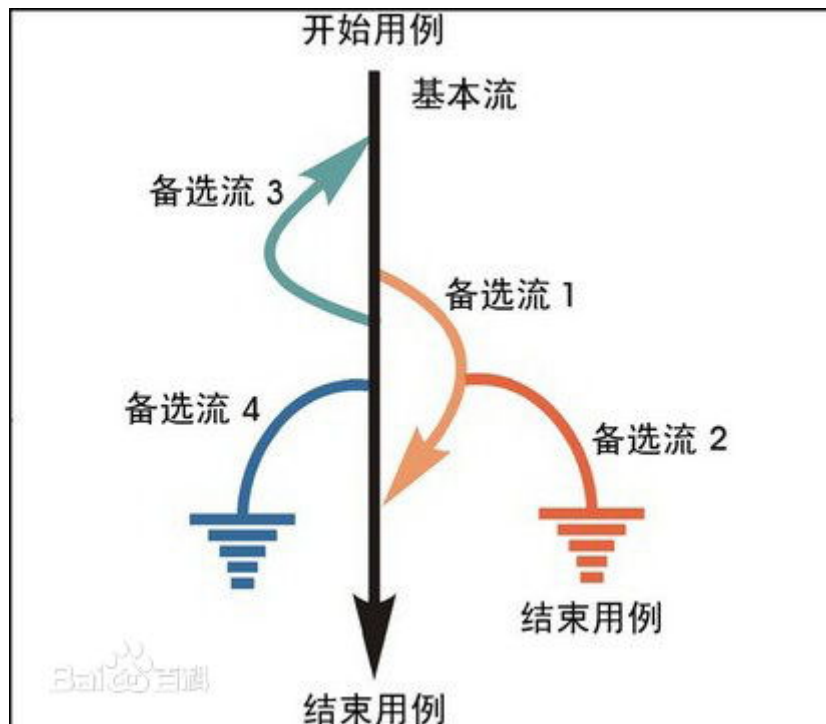
以注册为例

比兴PDF万兴PDF



想象注册的场景来设计用例，这与根据需求的业务流来设计差不多。主要是想象各种业务流来设计用例。例如我们可以再想象以下场景：

- 1、用户激活后再次点击邮件激活链接？
- 2、已注册用户再次注册？



因果图

因果图是一种简化了的逻辑图，能直观地表明程序输入条件（原因）和输出动作（结果）之间的相互关系。因果图法是借助图形来设计测试用例的一种系统方法，特别适用于被测试程序具有多种输入条件、程序的输出又依赖于输入条件的各种情况。

因果图的需要掌握的基本知识

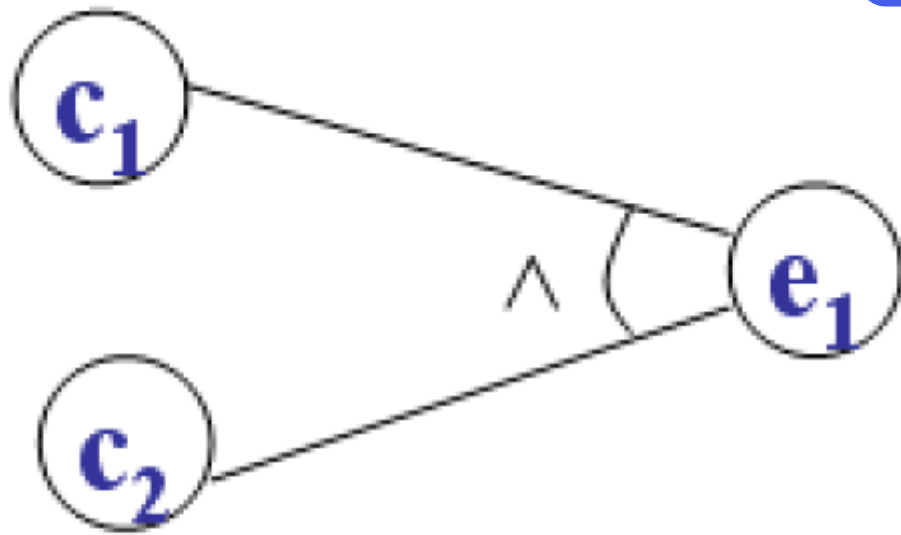
- 恒等



恒等

恒等：如果原因为真，那么结果必定为真。例如：动物园运来大熊猫，动物园一定有大熊猫

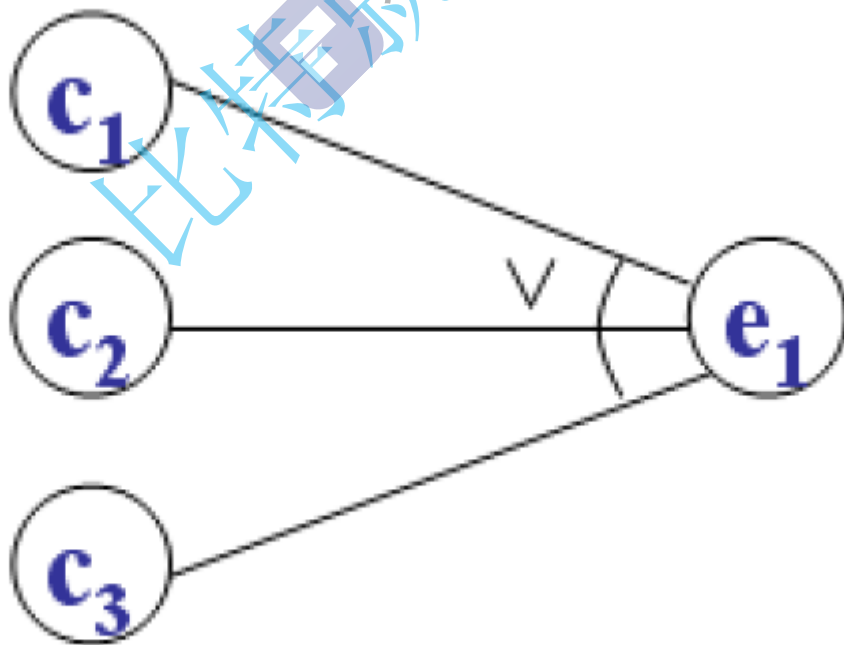
- 与



与

只有2个原因都为真，那么结果为真 例如：北京姑娘，必须有车且有房

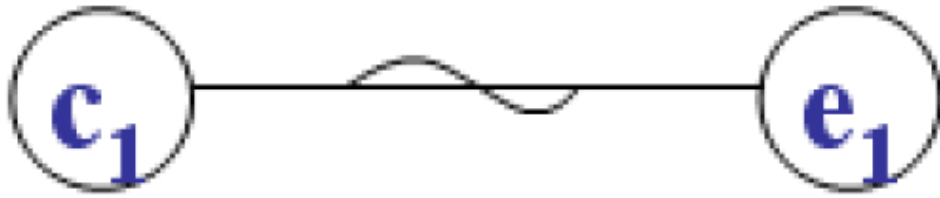
- 或



或

2个原因中有一个为真时，结果就为真。例如：长沙姑娘，你有车或者有房

- 非



非

只有原因为假，结果才为真。例如：你不好好学习，找到好工作

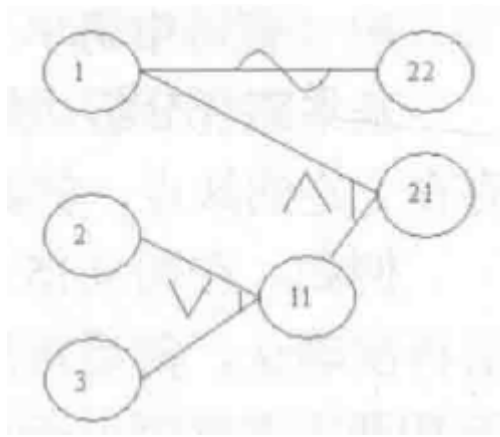
因果图法设计测试用例的步骤如下。

- (1) 分析所有可能的输入和可能的输出。
- (2) 找出输入与输出之间的对应关系。
- (3) 画出因果图。
- (4) 把因果图转换成判定表。
- (5) 把判定表对应到每一个测试用例。

案例一：

假设业务单据的处理规则为：“淘宝618活动，订单已提交，订单合计金额大于300元或有红包，则进优惠”。

1. 对于这条业务规则，首先通过分析所有可能的输入和可能的输出，可以得到如下结果：
 - 输入：订单已提交、金额大于300、有红包。
 - 输出：优惠、不优惠。
2. 然后，进行第二步，找出输入与输出之间的对应关系。通过分析，可以看出有以下的对应关系。
 - (1) 订单已提交，订单金额大于300元，则优惠。
 - (2) 订单已提交，订单金额小于等于300元，无红包，不优惠
 - (3) 订单已提交，有红包，则优惠。
 - (4) 订单已提交，订单金额大于300元，有红包，则优惠。
 - (5) 订单未提交，不优惠。
3. 为了方便画出因果图和判定表，需要对所有输入和输出编号，现在编号如下。
 - 1：订单已提交。
 - 2：订单金额大于300元。
 - 3：有红包
 - 21：优惠
 - 22：不优惠
4. 画因果图



5. 画判定表：有3个条件,输出有2个取值，所以表的列数为 $2 \times 2 \times 2 = 8$

		1	2	3	4	5	6	7	8
条 件	1	Y	Y	Y	Y	N	N	N	N
	2	Y	Y	N	N	Y	Y	N	N
	3	Y	N	Y	N	Y	N	Y	N
中 间 结 果	11	Y	Y	Y	N	Y	Y	Y	N
动 作	21	Y	Y	Y	N	N	N	N	N
	22	N	N	N	Y	Y	Y	Y	Y

6. 最终的测试用例

1, 2, 3, 4, 5 (包含6, 7, 8)。

案例二：

继续以注册的需求为例：

姓名、邮箱、密码、确认密码、验证码必须全部输入，才能进行注册

需要设计多少用例？ $2 \times 2 \times 2 \times 2 \times 2$ 。

因果法设计测试用例可以帮助测试人员理清输入和输出的关系，但是对于比较复杂的输入和输出，会耗费大量时间

正交排列

因果法设计用例太多怎么办？

正交法的目的是为了减少用例数目。用尽量少的用例覆盖输入的两两组合。

正交试验设计(Orthogonal experimental design)是研究多因素多水平的一种设计方法，它是根据正交性，由试验因素的全部水平组合中挑选出部分有代表性的点进行试验，通过对这部分试验结果的分析了解全面试验的情况，找出最优的水平组合。正交试验设计是一种基于正交表的、高效率、快速、经济的试验。

因素 (Factor)： 在一项试验中，凡欲考察的变量称为因素（变量）

水平 (位级) (Level)： 在试验范围内，因素被考察的值称为水平（变量的取值）

正交表的构成：

行数(Runs)： 正交表中的行的个数，即试验的次数,用N代表。

因素数(Factors)： 正交表中列的个数，用C代表。

水平数(Levels): 任何单个因素能够取得的值的最大个数。正交表中的包含的值为从0到数与水平数一致。从1到“水平数”，用T代表。

正交表的表示形式: $L = \text{行数}(\text{水平数} \times \text{因素数}) \quad L = N(TC)$

正交表的两条性质:

每一列中各数字出现的次数都一样多。

任何两列中的各有序数对出现的次数都一样多。

正交法设计测试用例的步骤:

- 1、有哪些因素 (变量)
- 2、每个因素有哪几个水平 (变量的取值)
- 3、选择一个合适的正交表
- 4、把变量的值映射到表中
- 5、把每一行的各因素水平的组合作为一个测试用例
- 6、加上你认为可疑且没有在表中出现的用例组合

案例:

继续以注册为例 (类似工具可以使用微软的PICT工具):

- 1、因素: 姓名、邮箱、密码、确认密码、验证码
- 2、水平: 填写、不填写

实验说明	选择正交表	因素与水平			
序号	1	2	3	4	5
因素名称	姓名	EMAIL	密码	确认密码	验证码
水平1	填写	填写	填写	填写	填写
水平2	不填写	不填写	不填写	不填写	不填写

3、表中的因素数=5;

表中至每个因素数的水平数=2

行数取最少的一个, 即试验次数最少的一个

$$L = N(TC) = (2-1) \times 5 + 1 = 6(25) \quad N = C \times (T-1) + 1$$

$$L = 6(25)$$

N试验次数

T水平数

C因素数

选择正交表, 这里选择了L6_2_5。正交表不是随便选择的, 而是设计好的

4、生成测试用例

思路: 因素取值为填写时: 正交按取值个数5-3-2-1 (5已全了, 3, 2, 1任意排列) 进行排列, 实验次数不够用取值为填写个数为2或3任意组合, 但要满足正交的二条性质。

实验计划表						
所在列	1	2	3	4	5	6
因素	姓名	email	密码	确认密码	验证码	实验结果
实验1	填写	填写	填写	填写	填写	
实验2	填写	填写	不填写	填写	不填写	
实验3	填写	不填写	填写	不填写	不填写	
实验4	不填写	填写	不填写	填写	不填写	
实验5	不填写	不填写	填写	不填写	填写	
实验6	不填写	不填写	不填写	不填写	填写	

5、增补测试用例

姓名、邮箱、密码、确认密码、验证码都不填写

发散一下：实际工作中这种做法会不会常见？在那种项目中需要这样做？实际中我们会怎么做？

不常见

科学实验性更需要

实际中可能会看代码或者想象可能的代码来减少用例

查询或添加页，有多个输入，如何测试？

什么样的测试用例是一个好的测试用例？

什么是测试用例的有效性？

测试用例对应的功能已删除，不可操作了

微信刚出来时与QQ可互发消息，下一个版本后就不可以发消息。

执行一条测试用例未发现BUG，实际该处有BUG

苹果7手机微信添加了mobile单车小程序，扫码不能开锁，只能使用mobile APP开锁，测试用例未涉及到苹果7微信小程序扫码开锁。

执行一条测试用例发现了BUG

苹果7手机微信添加了mobile单车小程序，用例已写到了苹果7微信添加mobile小程序扫码开锁，问题被发现

执行一条测试用例未发现BUG，实际该处BUG已修改

苹果7手机微信添加了mobile单车小程序扫码开锁，可以正常开锁

测试用例的粒度和评价

###测试用例的粒度

好的测试用例是一个不熟悉业务的人也能依据用例来很快的进行测试

粒度：指测试用例编写的详细程度。

测试用例可以写得很简单，也可以写得很复杂。最简单的测试用例是测试的纲要，仅仅指出要测试的内容，如探索性测试中的测试设计，仅会指出需要测试产品的哪些要素、需要达到的质量目标、需要使用的测试方法等。而最复杂的测试用例就像飞机维修人员使用的工作指令卡一样，会指定输入的每项数据，期待的结果及检验的方法，具体到界面元素的操作步骤，指定测试的方法和工具等。

(1)测试用例写得过于复杂或详细，会带来两个问题：一个是效率问题，另一个是维护成本问题。另外，测试用例设计得过于详细，留给测试执行人员的思考空间就比较少，容易限制测试人员的思维。

(2)测试用例写得过于简单，则可能失去了测试用例的意义。过于简单的测试用例设计其实并没有进行“设计”，只是把需要测试的功能模块记录下来而已，它的作用仅仅是在测试过程中作为一个简单的测试计划，提醒测试人员测试的主要功能包括哪些而已。测试用例的设计的本质应该是在设计的过程中理解需求，检验需求，并把对软件系统的测试方法的思路记录下来，以便指导将来的测试。

大多数测试团队编写的测试用例的粒度介于两者之间。而如何把握好粒度是测试用例设计的关键，也将影响测试用例设计的效率和效果。应该根据项目的实际情况、测试资源情况来决定设计出怎样粒度的测试用例。

主要考虑可以参考如下内容：

- 产品的质量要求
- 项目对用例的要求
- 测试时间和资源是否充分

但是不管用例怎么简化，都不应该省略

###测试用例的评价

测试用例设计出来了，如何提高测试用例设计的质量？就像软件产品需要通过各种手段来保证质量一样，测试用例的质量保证也需要综合使用各种手段和方法。评审分为正式和非正式评审。

- 同行评审
- 用户检查
- 项目组评审

(1) 测试用例的检查可以有多种方式 但是最敏捷的应当属临时的同行评审。同行评审，尤其是临时的同行评审，应该演变成类似结对编程一样的方式。从而体现敏捷的“个体和交互比过程和工具更有价值”，要强调测试用例设计者之间的思想碰撞，通过讨论、协作来完成测试用例的设计，原因很简单，测试用例的目的是尽可能全面地覆盖需求，而测试人员总会存在某方面的思维缺陷，一个人的思维总是存在局限性。因此需要一起设计测试用例。

(2) 除了同行评审，还应该尽量引入用户参与到测试用例的设计中来，让用户参与评审，从而体现敏捷的“顾客的协作比合同谈判更有价值”这一原则。这里顾客的含义比较广泛，关键在于如何定义测试，如果测试是对产品的批判，则顾客应该指最终用户或顾客代表（在内部可以是市场人员或领域专家）；如果测试是被定义为对开发提供帮助和支持，那么顾客显然就是程序员了。

(3) 由测试负责人组织协调开展会议，用例编写人对用例进行讲解，参会人员有异议的当场提出。

面试案例

某公司招聘测试工程师时，有一道这样的笔试题：“某手机软件有用TF卡导出数据的功能，请写出测试此功能点的思路”



考虑方向	检查点	测试思路描述
正向	导出数据正确性	导出数据，验证数据正确性
逆向	导出数据有效性	无数据时，导出功能是否正确
边界容量	TF卡空间不足	只能容纳部分数据
边界容量	TF卡容量已满	
容错	TF卡写保护	
容错	TF卡无法识别	
容错	人为中断	导出时拔掉TF卡
容错	导出时断电、关机等	再开机后检查能否正确导出
性能	连续多次导出	脚本实现，大量导出，查看数据是否正确
性能	检查导出速度	
兼容性	不同品牌和容量	
兼容性	不同分区格式FAT,FAT32, NTFS	不使用手机自带的T卡格式化功能