



测试设计

深圳市门道信息咨询有限公司 Shenzhen MT Information Consulting Co., LTD 版权所有. 侵权必究

目录



Chapter 1 测试用例设计规范

Chapter 2 黑盒用例设计技术

Chapter 3 设计方法综合策略

Chapter 1 测试用例设计规范



- 1.1 什么是测试用例
- 1.2 测试用例包含的要素
- 1.3 测试用例 参考
- 1.4 测试用例误区

1.1 什么是测试用例



- 测试用例是一份测试文档,其目的是确定系统的某个特性是 否正常工作。
- 测试用例是软件测试团队的主要工作成果之一。
- 测试用例的质量与写该用例的测试人员的水平关系极大。
- 执行测试用例是将这些用例逐个在被测的软件上执行,并判断其结果是否和预期相符。

1.2 测试用例包含的要素



- ■用例编号
- 模块
- 功能(用例属性)
- 用例标题
- 优先级(用例等级)
- 预置条件(前置条件)
- ■步骤
- 预期结果
- 编写者
- ■实际结果
- 备注

1.2.1 用例编号



- 1、用例编号:给用例定一个具体编号,用以唯一标识一条用例。
- 2、编号要唯一
- 3、技巧:
 - ① 项目名_模块名_编号
 - ② 模块名_编号
 - ③ 编号

1.2.2 模块/功能



- 1、模块:该用例所属模块名,一个模块下有一个或多个功能。 某些大模块还分子模块,具体分法根据项目业务和测试用例 的组织来确定,一般没有严格的规定。
- 2、功能:该用例所涉及的功能,每个功能下有一条或多条用例。

1.2.3 用例标题



- 1、测试标题:有的公司也叫测试目的。
- 2、标题不能重复。
- 3、标题是用例评审时的重要评审点,因此要不断练习,语言精炼。

1.2.4 优先级(讨论)



1、优先级:一般分为高、中、低

高:核心流程、冒烟用例

中:一般流程、异常流程

低: 界面、兼容

注意:不同的公司会有不同的优先级标识,如:1、2、3

1.2.5 预置条件



1、预置条件:一般不填写,除用例必须在特殊情况,特殊条件下才能执行时填写。

■ 不需要填写:

- ① 需要登录后才能点击某个连接或进入某个界面。
- ② 需要准备一个正确数据才能登录。
- ③ 需要添加数据才能执行查询。

■ 需要填写:

- ① 需要某种特定网络环境。
- ② 需要有某些权限才能执行用例。
- ③ 需要在某个用例执行后才执行本用例。

1.2.6 步骤



- 1、测试步骤: 描述具体如何操作的过程
- 2、执行人会根据步骤执行,因此编写后一定要有可执行性, 即执行人拿到后不会因为读不懂或看不明白而问用例设 计者。
- 3、一般包含:
 - ① 进入页面步骤,即路径
 - ② 输入了哪些数据
 - ③ 执行了哪些操作

1.2.7 期望结果



- 1、期望结果:按照测试步骤执行后,期望得到一个什么输出或者结果。
- 2、有的公司一对一,有的公司多对一。 如:一个步骤一个预期结果;多个步骤一个预期结果。

1.2.8 编写者



1、编写者:即编写该条用例的测试人员。

1.2.9 测试结果



- 1、测试结果: 执行该条用例的结果。
- 2、一般填写"通过"、"不通过"; 或"成功"、"失败"; 或"P"、"F"。
- 3、若未执行的用例,则可填写"未执行"或"TBC"。

1.2.10 备注



- 1、备注:用例需要做特殊说明时使用。
- 2、一般不填写任何内容。

1.3 测试用例 - 参考



CaseII	模块	功能	用例标题		预置 条件	输入步骤	预期输出
1	销售系统/销售计划	计划年度	计划年度超过本 年,新增一笔销 售计划单	高		1、【销售系统》销售计划】 2、计划类型:公司 计划年度:2020 3、输入或选择其他信息,保 证信息完整准确 4、点击"保存新增"	提示:销售计划新增成功等友好信息
2	销售系统/销售 计划	计划年度	计划年度输入为非纯数字	高		1、【销售系统》销售计划】 2、计划类型:公司 计划年度:20!16!! 3、输入或选择其他信息,保 证信息完整准确 4、点击"保存新增"	提示:"计划年度信息填写有误,请重新填写"等友好提示

1.3 测试用例 - 参考



CaseID	模块	功能	用例标题	优先 级	预置 条件	输入步骤	预期输出
3		销售计划反	未审核的销售 报价单被引用 后可以进行反 审核			1、点击"销售计划",按照要求填写完整正确的销售计划单,点击"保存新增" 2、【销售计划>>上一单】,使用有权限帐号对该销售计划单进行"审核"操作,点击"确定" 3、【销售报价>>操作>>引用销售计划单】,选中销售计划单,点击"确定",信息填写完整后点击"保存新增" 4、【销售报价》找到单据】,确认状态为"未审核" 5、【销售计划>>上一步】,找到被引用的销售计划单 6、点击"反审核"	
4		销售计划反	被己审核的销 售报价单被引 用后不可以进 行反审核			1、点击"销售计划",按照要求填写完整正确的销售计划单,点击"保存新增" 2、【销售计划>>上一单】,使用有权限帐号对该销售计划单进行"审核"操作,点击"确定" 3、【销售报价>>操作>>引用销售计划单】,选中销售计划单,点击"确定",信息填写完整后点击"保存新增" 4、【销售报价》找到单据》审核】,确认状态为"已审核" 5、【销售计划>>上一步】,找到被引用的销售计划单 6、点击"反审核"	提示:该单据 已被引用,不 能反审核等友 好信息

1.3 测试用例 - 练习



- 登录成功
- 登录失败
- 购买商品流程
- 由于库存不足,购买商品失败流程

1.4 测试用例误区



- ■实际结果不属于测试用例的组成部分。
- 用例由于条件不足,数据不全,不具备测试等原因,在填写 执行结果时除了通过和不通过,还有一个状态:未执行。
- 上述元素仅是用例公有部分,实际工作中各公司用例模板上 会有差异,如:有的公司还有额外一些字段(环境、URL、 开发者、参考资料等)。

Chapter 2 黑盒用例设计技术



- 2.1 黑盒用例设计技术
- 2.2 设计原因
- 2.3 等价类
- 2.4 边界值分析法
- 2.5 错误推测法
- 2.6 场景法
- 2.7 因果图法
- 2.8 判定表
- 2.9 正交试验

2.1 黑盒用例设计技术



- 等价类(重点)
- 边界值(重点)
- 错误推测法(重点)
- 场景法(重点)
- 因果图(了解)
- 判定表(了解)
- 正交试验(了解)

2.2 设计原因



- 为什么要设计出如此多的方法? 当我们测试一个输入框,条件是:6-20位数字、字母、下划 线时,如果把所有满足/不满足的情况全部测试一遍(穷 举),那么测几年都不够,面对一个系统,时间就更不可取, 因此我们需要一个切实可行的办法把穷举变得可测。
- 数学给了我们巨大帮助,其中集合就是重要的方法。



■ 概念

等价类是把所有可能的输入数据,即程序的输入域划分成若干部分(子集),然后从每一个子集中选取少数具有代表性的数据作为测试用例。



■ 加法器程序需求

输入两个两位整数,相加后,结果正确显示。要求:

- 1. 输入的整数值在-99和99之间(包含-99和99)
- 2. 非法输入应被拒绝,并显示错误信息



■ 如何设计加法器程序的测试用例

- 测试了 1+1、1+2、1+ (-1) 和 1+ (-2) 之后,是否有必要测试 1+3、1+4·····?
- 2、 不对加法器进行穷举测试,能否放心地认为其他的参数组合都是正确的? ₽

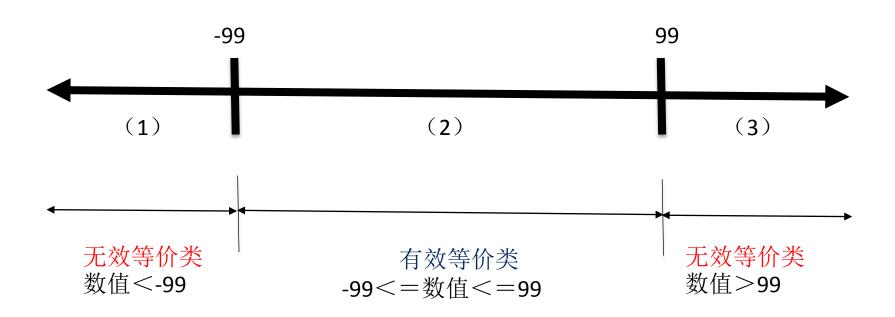


■ 等价类划分方法

- ① 一种最为典型的黑盒测试方法。₽
- ②把程序的输入划分成若干个部分,从每个部分中选取少数代表 性数据作为测试用例。。
- ③每一类的代表性数据在测试中的作用等价于这一类中的其他 值。↓



■ 针对加法器程序的输入划分等价类





首先把等价类划为有效等价类和无效等价类两个部分

■ 有效等价类

- ① 合理的、有意义的、接收后按照正常流程执行的数据所构成的集合。
- ② 利用有效等价类可检验是否实现了需求所规定的功能。

■ 无效等价类

- ① 不合理的、没有意义的、接收后按照非法流程执行或无法继续执行的数据所构成的集合。
- ② 利用无效等价类可检验在异常状态下程序是否给出了友好提示或作出了相应判断。



■ 每一个等价类分配一个唯一的编号

序号₽	功能项。	有效等价类。	编号□	无效等 价类 。	编号	تها
1₽	- 	00~神楽班(本~ 00 -	2₽	加数取值<-99₽	1₽	Ţ
	加法器』 -99	-99≤加数取值≤ 99₽		加数取值>99。	3₽	Ę



■ 设计测试用例

- 1、设计一个测试用例,使它能够尽量覆盖尚未覆盖的有效等价类。重复该操作,从而使所有有效等价类均被覆盖。
- 2、设计一个测试用例,使它能够覆盖一个无效等价类。重复该操作,从而使所有无效等价类均被覆盖。

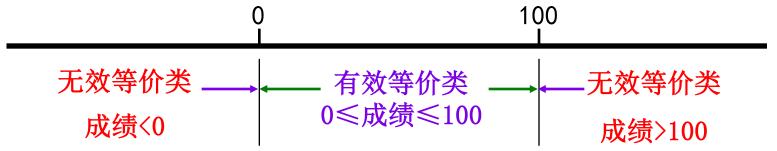
测试用例编号。	输入数值。	所属等价类。	预期输出₽
1₽	-50+24₽	2₽	正确输出: -26-
2₽	-130+77₽	1₽	错误信息₽
3₽	-9+125₽	3₽	错误信息₽

2.3 等价类划分6条细则



1、 在输入条件规定了取值范围或值的个数的情况下,则可以确立一个有效等价 类和两个无效等价类。

如:学生成绩区间0~100,那么要选取小于0,0到100之间,大于100



- 2、在输入条件规定了输入值的集合或者规定了"必须如何"的条件的情况下, 可确立一个有效等价类和一个无效等价类。
 - 如: 姓名必须输入中文, 那么要选取中文一次, 非中文一次
- 3、 在输入条件是一个布尔量的情况下,可确定一个有效等价类和一个无效等价类。

如:录入性别,仅能是"男"或"女"。 要求输入"男",则"男"是有效等价类;"女"就是无效等价类; 要求输入"女",则"女"是有效等价类;"男"就是无效等价类。

2.3 等价类划分6条细则



4、在规定了输入数据的一组值(假定n个),并且程序要对每一个输入值分别处理的情况下,可确立n个有效等价类和一个无效等价类

如:录入学历,可为:大专、本科、硕士、博士4种之一

有效等价类:大专、本科、硕士、博士

无效等价类: 4种学历之外的任何学历

5、在规定了输入数据必须遵守的规则的情况下,可确立一个有效 等价类(符合规则)和若干个无效等价类(从不同角度违反 规则)

如: 录入年龄,年龄必须是数字。

有效等价类:数字

无效等价类: 字母、汉字、符号

6、在确知已划分的等价类中各元素在程序处理中的方式不同的情况下,则应再将该等价类进一步的划分为更小的等价类 综合考虑

2.3 等价类设计用例方法



- 1、划分为有效和无效两个部分
- 2、列出所有的有效类和无效类
- 3、设计足够少的测试用例,使其足够多的覆盖有效类
- 4、设计足够多的测试用例,使其一一覆盖无效类

2.3 等价类划分实例



例1: 某程序规定: "输入三个整数 a、b、c分别作为三边的边长构成三角形。通过程序判定所构成的三角形的类型,当此三角形为一般三角形、等腰三角形及等边三角形时,分别作计算…"。用等价类划分方法为该程序进行测试用例设计。

2.3 等价类划分实例



分析题目中给出和隐含的对输入条件的要求:

- (1) 整数 (2) 三个数 (3) 非零数
- (4) 正数 (5) 两边之和大于第三边
- (6) 等腰 (7) 等边

如果 a 、 b 、 c 满足条件(1)~(4),则输出下列四种情况之一:

- 1. 如果不满足条件(5),则程序输出为"非三角形"。
- 2. 如果三条边相等即满足条件(7),则程序输出为"等边三角 形"。
- 3. 如果只有两条边相等、即满足条件(6),则程序输出为"等 腰三角形"。
- 4. 如果三条边都不相等,则程序输出为"一般三角形"。

2.3 等价类划分实例



列出等价类表并编号

	I	- 1		T* 先后发生 (人 214	D 777		
		有效等价类	- 등	无效等价类	号码		
		<u> </u> 型	码				
				│ a 为非整数	12		
	輸			一边为非整数 6为非整数	13		
輸	-11 11	++	١.	│ c为非整数	14		
ניודר		整数	1	⊢a,b 为非整数	15		
	入			两边为非整数 b,c 为非整数	16		
	^			La,c 为非整数	17		
				三边 a,b,c均为非整数	18		
入	三			┌ 只给 a	19		
				│ 只给一边 ┤ 只给 6	20		
		三个数		L 只给 。	21		
_	个		2	厂只给 ab	22		
条				只给两边 ~ 只给 b, c	23		
					24		
件	整			给出三个以上	25		
1 ''				∠a 为0	26		
	数			一边为零 🚽 6为0	27		
				-1L- -		└ 。为 0	28
		非零数	3		29		
				二边为零 { a,b 为 0 b,c 为 0 a,c 为 0	30		
					31		
				三边 a,b,c均为0	32		
				~ a<0	33		
				一边⊲ → 6⊲0	34		
		正数		L °<0	35		
			4	CaO且b⊲	36		
				二边勾一人。勾且。勾	37		
					38		
					39		
				三边均幻: 4幻且 6幻且 6幻			

2.3 等价类划分实例



列出等价类表并编号 (续上表)

输出条	构成一般 三角形	a+b>c b+c>a a+c>b	5 6 7	a+b<0 a+b=0 b+c <a b+c=a -{ a+c<b a+c=b</b </a 	40 41 42 43 44 45
件	构成等腰 三角形 构成等边 三角形	a=b b=c 且两边 之和 太于第 三边 a=b=c	8 9 10		

2.3 等价类划分实例



覆盖有效等价类的测试用例:

a b c 覆盖等价类号码

3 4 5 (1) -- (7)

4 4 5 (1) -- (7), (8)

4 5 5 (1) -- (7), (9)

5 4 5 (1) -- (7), (10)

4 4 4 (1) -- (7), (11)

2.3 等价类划分实例



覆盖无效等价类的测试用例

a	b	c	覆盖等价类号码	a	b	С	覆盖等价类号码
2.5	4	5	12	0	0	5	29
3	4.5	5	13	3	0	0	30
3.	4	5.5	14	0	4	0	31
3.5	4.5	5	15	0	0	0	32
3	4.5	5.5	16	-3	4	5	33
3.5	4	5.5	17	3	-4	5	34
4.5	4.5	5.5	18	3	4	-5	35
3			19	-3	-4	5	36
	4		20	-3	4	-5	37
		5	21	3	-4	-5	38
3	4		22	-3	-4	-5	39
	4	5	23	3	1	5	40
3		5	24	3	2	5	41
3	4	5	25	3	1	1	42
0	4	5	26	3	2	1	43
3	0	5	27	1	4	2	44
3	4	0	28	3	4	1	45

2.4 边界值分析法



- 边界值分析法是对输入或输出的边界值进行测试的一种黑盒 测试方法。
- 边界值分析方法是对等价类划分方法的补充,其测试用例来 自等价类的边界。
- 错误隐藏在角落里,问题聚集在边界上。

长期的测试工作经验告诉我们,大量的错误是发生在输入或输出范围的边界上,而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例,可以查出更多的错误。

■ 应当选取正好等于,刚刚大于或刚刚小于边界的值作为测试数据,而不是选取等价类中的典型值或任意值作为测试数据。

2.4 与等价划分的区别

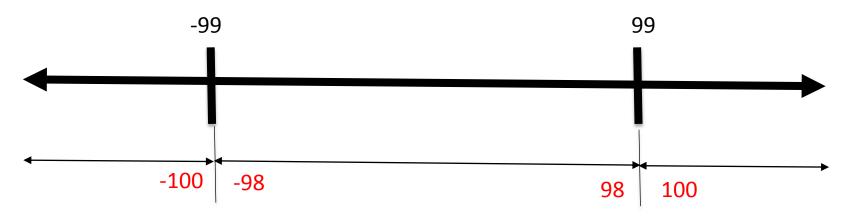


- 1、边界值分析不是从某等价类中随便挑一个作为代表,而是使 这个等价类的每个边界都要作为测试条件。
- 2、边界值分析不仅考虑输入条件,还要考虑输出空间产生的测试情况。

2.4 边界值



■ 针对加法器程序的输入边界值分析



测试用例编号₽	输入数值₽	测试边界₽	预期输出↩	4
143	-100+25↔	-99₽	错误提示↩	4
2₽	-99+ (-99) ↔		输出:-198₽	4
3₽	-98+ (-98) ¢		输出:-196₽	4
4₽	-11+100₽	994	错误提示₽	4
5₽	98+98₽		输出:196₽	4
6₽	99+99₽		输出:198₽	4

2.4 边界值细则



1、如果输入条件规定了值的范围,则应取刚达到这个范围的边界的值,以及刚刚超越这个范围边界的值作为测试输入数据。

如:输入范围是0-50,

则测试用例应该包含-1、0、1、49、50、51,

即正好等于、刚刚大于或刚刚小于边界的值。

2、如果输入条件规定了值的个数,则用最大个数,最小个数, 比最小个数少一,比最大个数多一的数作为测试数据。

如:课程名称最多输入12个字符,

则应该设计输入11个字符、12个字符、13个字符作为测试用例。

2.4 边界值细则



3、如果程序的规格说明给出的输入域或输出域是有序集合,则 应选取集合的第一个元素和最后一个元素作为测试用例。

如:查询95届-98届学生的各科绩。

则应该设计测试用例包括95届和98届,还应查询94届、99届学生成绩的用例。

另,查询显示,要求最少显示1条记录,最多显示10条,则应设计测试用例包括显示1和10,还应包括显示0和11。

4、如果程序中使用了一个内部数据结构,则应当选择这个内部数据结构的边界上的值作为测试用例。

如:程序中定义了一个数组,其元素下标的下界是0、上界是100,那么应选择达到这个数组下标边界的值,如0和100作为测试用例。

2.4 边界值 一 图片上传



■ 大小不超过5M

添加

■ 为上传图片添加水印 水印版权须知

提示: 1.选择的文件大小不超过5M,支持jpg、gif、png;

- 2. 只能为尺寸大于300×300的jpg文件添加水印。
- 3. 使用mac操作系统的网友,如您在上传过程中遇到问题,请重

新安裝flash后再试,免费下载。



思考下: 都需要选取哪些值?

2.4 边界值 - 图片上传



- 5M(正好等于)
- 4.9M (略小于最大值)
- 5.1M (刚刚大于)
- 3M (正常值)
- 0.1M(略高于最小值)
- 10M (远大于最大值)

2.5 错误推测法



- 错误推测法: 基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法。
- 错误推测方法的基本思想: 列举出程序中所有可能有的错误和容易发生错误的特殊情况,根据他们选择测试用例。

2.5 错误推测法



- 例如: 一个密码输入框
 - (输入错误的数据,应有相对应的提示。)
 - 1、空
 - 2、混合
 - 3、左右空格
 - 4、2条数据完全一致(在不允许一致的情况下)
 - 5、极限

大家思考一下"ATM取钱"?



- 前面的讨论侧重于数据的选择,不涉及操作步骤,无法对涉及到用户操作的过程进行全面测试。
- 对于复杂的系统,我们更应该侧重整体的业务流程,确保每个功能的各个分支和异常得到充分的测试。
- 我们系统的每个流程都看成是为了完成实际工作的一个场景 (业务),因此场景法就诞生。

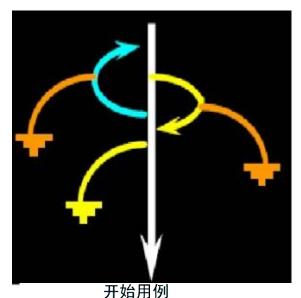
2.6 场景法



- 场景法也叫流程分析法,是将软件系统的某个流程看成路径, 用路径分析的方法来设计测试用例。
- 根据流程的顺序依次进行组合,使得流程的各个分支都能走到。

2.6 场景法





基本流备选流 1备选流 2 4束用例

- 事件流组成
 - 基本流
 - 仅有一个基本流,白色箭头
 - 是经过用例的最简单的路径, 指每个步骤都"正常"运作 所行进的路线。
 - 备选流
 - 可以有多个
 - 可选或备选情况
 - 异常事件

2.6 场景法



■ 基本流和备选流的区别

	基本流	备选流
测试重要性	重要	次要
数目	1条	1条或多条
初始节点	初始状态	基本流或其他备选流
终止节点	终止状态	基本流或者其他终止状态
业务流程完整性	完整	不完整, 仅为片段
能否构成场景	能	不能,需要基本流

2.6 场景法设计步骤



- 1、分析被测业务,找出基本流及各项备选流。
- 2、根据基本流和各项备选流生成不同的场景。
- 3、对每一个场景生成相应的测试用例。



■ ATM取款





■ 备选流

- 1、插卡
 - ① 卡插反了
 - ② 不是银行卡
 - ③ 消磁卡
 - ④ 黑名单卡
 - ⑤ 已冻结卡
 - ⑥ 己注销卡

2、输密码

- ① 1次错误
- ② 2次错误
- ③ 3次错误



- 备选流
 - 3、选择"取款"
 - ① 选择其他业务
 - 4、输入金额
 - ① 规则: 单笔100倍, 单笔最高5000, 单日最多2W
 - ② 根据这个规则,继续设计一堆用例
 - 5、验ATM
 - ① ATM钱不足
 - ② ATM没钱



编号	模块	功能	用例标题	优先级	预置条件	输入步骤	预期输出	
1	取款	取款	信息完整,取款成功	恒		1、插入磁卡 2、输入密码,如: 123456 3、点击"取款" 4、输入取款金额: 100 5、点击"确定"	取款成功	
2	取款	取款	将磁卡反着插,取款失败	恒		1、反向插入磁卡	5秒后退出磁卡	

2.7 因果图法



- 前面介绍的等价类划分方法和边界值分析方法都是着重考虑输入条件,但未考虑输入条件之间的联系,相互组合等。
- 这就需要利用因果图方法将所有的组合进行过滤后设计用例。
- 实际工作中,因果图直接转判定表,这种方法实用性不大。

参考资料

因果图(点击图标 直接打开文档)

2.8 判定表



- ■前面因果图方法中已经用到了判定表。
- 判定表(Decision Table)是分析和表达多逻辑条件下执行不同操作的方法。它可以把复杂的逻辑关系和多种条件组合的情况表达得既具体又明确。

		1	2	3	4	5	6	7	8
问	你觉得疲倦吗?	Y	Y	Y	Y	N	N	N	N
题	你对内容感兴趣吗?	Y	Y	N	N	Y	Y	N	N
	书中内容使你胡涂吗?	Y	N	Y	N	Y	N	Y	N
	请回到本章开头重读	X				X			
建	继续读下去		X				X		
议	跳到下一章去读							X	X
	停止阅读,请休息			X	X				

"读书指南"判定表

2.8 判定表



- 判定表的建立步骤:
 - 1、找出所有的因和果。
 - 2、确定因果存在的组合有多少种。假如有n个条件,每个条件有两个取值(0,1),故有2ⁿ种组合。
 - 3、列出所有组合,得到初始判定表。
 - 4、竖着执行,把初始判定表中不可能的情况全部去除,剩 下的就是最终判定表。

2.9 正交试验



- 利用因果图来设计测试用例时,作为输入条件的原因与输出结果之间的因果关系,有时很难从软件需求规格说明中得到。
- 往往因果关系非常庞大,以至于据此因果图而得到的测试用例数目多的惊人,给软件测试带来沉重的负担,为了有效地,合理地减少测试的工时与费用,可利用正交实验设计方法进行测试用例的设计。



参考资料 正交试验(点击图 标直接打开文档)

2.9 正交试验步骤



- 1、提取功能说明,构造因子--状态表 把影响实验指标的条件称为因子,而影响实验因子的条件叫 因子的状态。首先要根据被测试软件的规格说明书找出影响 其功能实现的操作对象和外部因素,把他们当作因子,而把 各个因子的取值当作状态。
- 2、加权筛选,生成因素分析表 对因子与状态的选择可按其重要程度分别加权,可根据各个 因子及状态的作用大小,出现频率的大小以及测试的需要,确定权值的大小。
- 3、利用正交表构造测试数据集 正交表的推导依据**Galois**理论(这里省略,需要时可查数理 统计方面的教材)。

Chapter 3 设计方法综合策略



- 3.1 设计方法如何选择
- 3.2 测试用例的粒度
- 3.3 测试用例的更新

3.1 设计方法如何选择



- ■当遇到需要输入数据时,首先进行等价类和边界值,这是减少工作量和提高测试效率最有效方法。
- ■用错误推测法再追加一些测试用例。
- ■对照程序逻辑,如果存在组合,用判定表设计用例。
- ■对于业务流清晰的功能,利用场景法贯穿整个测试案例过程,再针对不同节点综合使用其他测试方法。
- ■测试用例的设计方法不是单独存在的。
- 在实际测试中,往往是综合使用各种方法才能有效提高测试 效率和测试覆盖度。

3.2 测试用例的粒度



- 1、粒度,指的是粗细程度。粒度大,就是说一个用例所涵盖的 关注内容比较多,反之同理。
- 2、用例的粒度大,则总的用例数就少,用例看起来也简洁。
- 3、用例的粒度小,则单条用例关注的测试点很集中,不容易遗漏,并且执行需要的时间比较好估计。

■实际过程中是根据测试时间、项目阶段、主观/客观因素综合决定的。

3.3 测试用例的更新



■测试用例并不可能一开始就写得很完美,可能也有写错的,可能也有遗漏的测试点。

■随着软件的版本不断更新,软件本身的需求和规格以及设计都可能在不断的变更。

■随着测试的不断开展,测试人员对产品的理解逐渐加深。

联系我们



电话: 0755-83221336/13928429246 (微信同步)

邮箱: <u>service@mtesting.net</u>

官网: www. mtesting. net

学习社区: www. mtesting. cn

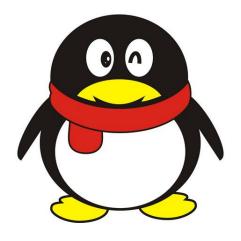
地址: 深圳市福田区彩田南路深圳青年大厦五楼



手机扫描访问官网



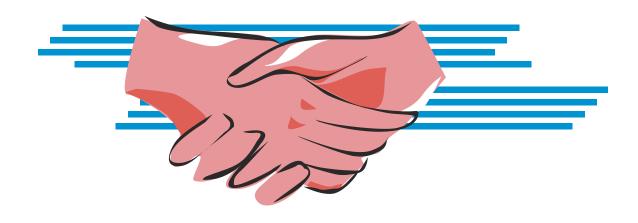
手机扫描关注公众号



QQ群号:15233368

感谢您对我们的关注





Thanks&Best wishes for you!

