测试用例

1. 基本概念
2. 测试用例（案例）主要记录：测试步骤、方法、数据、预期结果的文档，由测试人员在执行测试之前编写的。
3. 编写测试用例的方法：

1）等价类划分（数据）

2）边界值（数据）

3）因果图（逻辑关系）

4）判定表（条件项）

5）正交排列法（取值个数底，因子幂）

6）场景法

7）错误推测

8）功能图

3、 写用例参考什么？

1）文档：需求、开发文档、用户手册

2）参考已经开发出来的软件

3）讨论

1. 等价类划分
2. 应用场合

——只要有数据输入的地方，就可以使用等价类划分

——把无限多的数据根据需求，划分成多个区域（有效、无效），从每个区域中选取一个代表性数据进行测试即可

1. 说明：穷举测试是最全面的测试，但是是不能采用的方法，时间成本太高，编写用例的方法主要解决的问题是如何使用最少的数据，达到最大的覆盖
2. 核心概念
3. 有效等价类：对程序规格有效的、合理的输入数据的集合；程序接收到有效等价类，可以正确计算、执行
4. 无效等价类：对程序规格无效的、不合理的输入数据的集合

4、如何使用？

——首先明确测试对象—第一个数文本框

说明：在测试第一个数的时候，保证第二个数正确

1. 根据需求，划分等价类

①有效等价类

-99—99之间的整数

②无效等价类

1. 非整数
2. <-99的整数
3. >99的整数

（2） 细化等价类

——往往依据的不是字面的需求，而是基于对数据存储方式的深入理解以及数据格式的理解

1. 正负数补码计算不一样，有必要把正数、负数单独测试
2. -99—0整数
3. 0—99整数

② 非整数可以进一步细分

1. 小数
2. 字母
3. 汉字

D、 符号

（3）建立等价类表（熟练后直接做该步）

|  |  |  |
| --- | --- | --- |
| 有效等价类 | | |
| 编号 | 数据要求 | |
| 1 | -99—0整数 | |
| 2 | 0—99整数 | |
|  | | |
| 无效等价类 | | |
| 编号 | | 数据要求 |
| 1 | | <-99 |
| 2 | | >99 |
| 3 | | 小数 |
| 4 | | 字母 |
| 5 | | 汉字 |
| 6 | | 符号 |
| 7 | | 为空 |

说明：进行需求（数据）分析，是最核心的步骤，决定测试的整体质量

（4） 编写用例

——从每个等价类中，至少选择一个数据进行测试

说明：对于无效等价类，开始的时候最好每次只测试一个，最后再适当考虑无效等价类的组合；对“第二个数”文本框测试，保证第一个数正确

总结：

一个好的软件一般要满足两个标准：

（1）完成指定的功能——使用有效等价类

（2）有很强的容错能力（健壮性）——使用无效等价类

三、 边界值

1、 应用场合

只要有数据输入的地方，一般就可以使用边界值

一般情况下等价类和边界值共同使用，形成一套较为完善的方案

2、 如何使用

找到有效数据和无效数据的分界点（最小值、最大值），对该分界点及其两边的点，单独进行测试

3、 说明

边界值的数据，本质上属于等价类的范畴，但是需要单独进行测试，这种冗余在工程中是必要的。

四、 用例的优化

1、不同控件的有效等价类或有效边界值，可以尽可能多的在一条用例中测试——不同控件的有效等价类可以组合——减少用例数量

2、在一条用例中，开始的时候，只测试一个控件的一个无效等价类或无效边界值——无效等价类开始的时候不能组合——避免屏蔽现象发生（前面的错误提示出现后，后面控件的错误提示就不出现了）

最后再适当考虑不同控件和同一控件的多个无效等价类的组合——验证一下极端情况系统不崩溃

五、无效等价类重点考虑

1、必填项（不能为空）——为空

2、不能重复——重复

3、数据要求范围——超出范围

如：年龄：18—60，测试<18,>60

4、数据有字符个数要求——超出范围

如：姓名：1-20字符，测试<1字符，>20字符

5、从数据允许的样式考虑——小数、字母、符号、汉字

6、小数——小数点保留位数不满足

第十三集

因果图+判定表

一、 应用场合

在一个界面中，有多个输入条件，输入之间存在组合关系，不同的输入组合会产生不同的输出结果的组合，为了弄清输入和输出的关系，使用因果图

二、核心概念

1、因——输入条件

2、果——输出结果

使用画图的方法找出输入（因）和输出（果）的关系

三、图形符号

1、基本符号——表示输入和输出之间的关系

2、约束（限制）符号——限制的是同一类型的（同时限制的是输入或者同时限制的是输出）

说明：

互斥和唯一的区别：

一般有默认选项—唯一

没有默认选项—互斥

四、分析程序

1、找出所有的输入（因），编号

（1）投币50元

（2）投币100元

（3）充值50元

（4）充值100元

2、找出所有的输出（果），编号

A、充值成功并退卡

B、提示充值成功

C、找零

D、错误提示，退卡

前两步就是初步分析需求的过程

3、在步骤1基础上，找出哪些输入不能组合（限制），哪些输入可以组合

（1）不能组合（限制）

①输入（1）（2）不能组合

②输入（3）（4）不能组合

（2）能组合（决定测试用例的数量）

①输入（1）（3）

②输入（1）（4）

③输入（2）（3）

④输入（2）（4）

⑤输入（1）单独

⑥输入（2）单独

⑦输入（3）单独

⑧输入（4）单独

先把输入的限制关系画出因果图

4、在步骤2基础上，找出哪些输出不能组合，哪些可以组合

（1）不能组合

①输出AD不能组合

②输出BD不能组合

（2）能组合

①输出AB必须组合

②输出ABC可能组合

③输出CD可能组合

④输出D单独

步骤3、4深入分析需求的过程

画出输出的限制关系

5、根据步骤3（2）和4（2），找出什么样的输入组合会产生什么样的输出组合

情况1：

输入（1）（3）组合

会产生：

输出AB组合

情况2：

输入（1）（4）

会产生：

输出CD组合

=================

情况3：

输入（2）（3）

会产生：

输出ABC组合

=================

情况4：

输入（2）（4）

会产生：

输出AB组合

===================

情况5：

输入（1）单独

会产生：

输出CD组合

==============

情况6：

输入（2）单独

会产生：

输出CD组合

================

情况7：

输入（3）单独

会产生：

输出D

================

================

情况8：

输入（4）单独

会产生：

输出D

6、 根据因果图，写出判定表

7、根据判定表，写用例

判定表的一列转换成一条用例

五、因果图应用限制

考虑控件的组合，但是每个控件的取值条件不宜过多，一般为2-3项，如：按钮、

单选按钮、复选框、取值只有2-3项的下拉列表

无效等价类处理方式：

1、允许输入无效等价类数据，后面给出错误提示

2、允许输入无效等价类数据，但可以自动纠正成正确数据

3、不允许输入无效等价类数据

第十七集

正交排列法

一、应用场合

有很多的控件，每个控件有多个取值，要考虑不同控件不同取值的组合——如何使用最少、最优化的数据组合达到最大的测试覆盖

和因果图法的区别：

因果图法适用于控件组合数量较少（20种以下），要比较全面的考虑所有情况（或主要情况）

正交排列法适用于控件组合数量庞大，而从代码角度讲又没有必要全部测试

二、正交表

三、如何使用

1、分析需求——列出所有的控件和取值

2、根据控件的个数和控件的取值，选择一个合适的正交表

（1）根据控件的个数，决定正交表的“次幂”（正交表的列数）

4个控件——4次幂（4列）

（2）根据每个控件的取值个数，决定正交表的“底”（正交表中允许出现的最大值）

每个控件有 3 个取值——底为3

3、把正交表中的列名和取值（1、2、3…）用控件名称和控件的真正取值替代

（1）把列名用控件名称替代

（2）把每列中的取值（1、2…）使用对应的控件真正的取值替代

4、根据最终正交表编写用例

把正交表的每一行转换成一条用例

说明：

这些组合是经过数学推理出来的最少、最优化的用例

如果时间允许，最好再补充一些用例

四、正交表的局限性

正交表个数有限，并基本要求控件取值个数相等

五、正交表的测试思想

1、公平原则：使每个控件的每个取值参与组合的次数尽量相同

2、均匀原则：在所有的组合中挑选数据时，应该均匀零星的选取，而不要只从某个局部选

六、案例2

每个控件的取值不相同

1、控件的个数决定次幂，如果没有，找一个最接近的（一般选大一点的）

4个控件——4次幂

2、如何选“底”

方案1：少数服从多数原则

——有更多的控件取值相同的

有 2 个控件取值为 3 ——选底为“3”

方案2：取值个数最多原则

——看哪个控件取值最多

最多有 4 个 值——底为“4”

第十九集6/11day08

综合场景法

一、应用场合

1、适合使用场景法软件界面特点：界面中有很少（或没有）填写项，所有的操作都是通过鼠标的单击、双击、拖拽等完成

2、把自己当成最终的用户，尽可能真实全面的模拟用户的操作，设计出相应的测试点，一般包括两类：

（1）模拟用户正确的操作、完成主要业务逻辑的动作——验证软件的主要功能是否实现

（2）模拟用户错误的操作——验证软件错误处理能力

3、场景法主要基于：

（1）业务（需求）层面：对所测软件的重要功能、业务逻辑、行业背景深入理解

（2）技术层面：基于等价类划分，有效等价类——模拟用户正确操作；无效等价类——模拟错误操作

二、核心概念

1、基本流（正确流、有效流）

模拟用户正确的操作流程

2、备选流（错误流、无效流）

模拟用户错误的操作流程

三、使用步骤

1、根据需求，列出基本流和备选流

（1）基本流——正确取款

（2）备选流——在取款过程中出现的主要错误

此步骤完全基于业务的理解

2、根据基本流和备选流，生成场景（熟练后，直接做该步）

3、根据场景，编写用例

场景和用例并不是一一对应的关系

练习一：五子棋

1、 列出主要场景

2、 执行测试，把测试过的场景留下证迹（截图）

练习2：

列出删除类型主要的场景（在测试中会遇到哪些情形），并编写用例

练习3：

列出“删除房间”主要的场景（在测试中会遇到哪些情形），并编写用例

单元测试

（1） 最小的测试单位

（2） 依据是详细设计文档

（3） 以功能测试为主，重点模块结合白盒测试（检查代码）

（4） 一般需要编写驱动模块或桩模块

驱动模块：模拟被测模块上一级模块（调用被测模块的那个模块）

桩模块：模拟被测模块下一级模块（被被测模块所调用的那个模块）

集成测试

拿到一个新的版本，首先进行冒烟测试：利用较少的人（1-3人，经验更丰富）、较少的时间（0.5—2天）对软件的主要功能进行整体的测试，决定该本版是否值得一测

思路：

（1） 冒烟测试

（2） 返测（对上一个版本中发现的bug进行验证，看是否解决）

（3） 回归测试（对上一个版本中的所有功能进行验证）

（4） 对增加的新功能进行测试

现在很多公司没有真正的单元测试和集成测试阶段——节省成本

验收测试（uat：user acceptance test用户体验测试）

（1）Alpha测试：在开发的环境中，由最终的用户（可能由开发方模拟）对产品进行测试

（2）beta测试：在用户实际使用环境中，由最终的用户对产品进行测试

公共类软件一般发布beta版（公测版）免费让用户使用，收集bug信息

软件测试模型：

1、表达软件测试过程和开发过程之间的关系

2、V模型

优点：测试阶段划分明确，既包含代码级测试（单元测试）又包括用户级测试（验收测试）

缺点：使人容易理解成测试只是开发以后的工作，不符合趁早测试和不断测试的原则

深入理解：

在开发过程中，测试人员需要参与文档测试，并根据相关的文档对后续的测试工作进行规划（写计划、写用例）

一个程序不论哪个阶段必须要做的是黑盒测试

白盒测试只是对高风险的核心模块进行测试

白盒测试成本高、效率低，主要应用在单元测试阶段

静态代码测试主要检查代码的标准和规范性

白盒测试主要检查代码的逻辑和结构，更关注于功能

把静态测试盒白盒测试结合在一起就是——静态白盒测试