东软机密

文件编号：D00-CDG010

测试设计指南

版本：1.1.0-0.0.0

2009-6-12

东软集团股份有限公司 过程改善中心

文件修改控制

修改编号	版本	修改条款及内容	修改日期
1	1.0.0-0.0.0	新文件建立。	2008-7-30
2	1.1.0-0.0.0	根据公司要求，转换为OpenOffice格式	2009-6-12

目　录

1. 目的

本文的目的是描述系统测试设计的基本方法，为测试设计活动提供支持。

2. 测试设计指南

2.1 测试设计概述

2.1.1 测试的目的

测试的目的决定了如何去组织测试，测试的目的不同，选择的方法也不同。传统上按测试的目的将软件测试的方法分为两类，第一类测试方法是试图验证软件是“工作的”，所谓“工作的”就是指软件的功能是按照预先的设计执行的；而第二类测试方法则是设法证明软件是“不工作的”。

在测试活动中，通常将两类方法结合起来，以第一类测试方法为基础和主要线索，阶段性地运用第二类测试方法。

2.1.2 测试用例的特点

测试用例是为特定的目的而设计的一组测试输入、执行条件和预期的结果。测试用例是执行的最小实体。简单地说，测试用例就是设计一个场景，使软件程序在这种场景下，必须能够正常运行并且达到程序所设计的执行结果。总之，测试用例承载着测试人员的测试思路。

一个优秀的测试用例应满足下列条件：

- 最有可能发现缺陷

测试的目的之一就是为了发现系统的缺陷，寻找测试用例的设计灵感，可以沿着“程序可能会怎样才能出现错误”这条思路进行设计。

- 不是重复的、多余的

这个条件使测试执行更加有效。

- 一组相似测试用例中最有效的

在相似用例的集合中，你需要找出该类中最佳的用例，即最有可能发现缺陷的一个。

- 既不是太简单，也不是太复杂

可以通过把两种以上的测试组合为同一个测试用例来节约测试时间，但要掌握好尺度，不要过于复杂无法执行或晦涩难懂，也不要使一个用例的测试时间过长。

测试用例能够代表并覆盖各种合理的和不合理的、合法的和非法的、边界的和越界的以及极限的输入数据、操作和环境设置，每一个测试用例都应有相应的唯一的期待结果。

2.2 测试设计方法

测试方法有很多种，此文中针对测试中经常使用的三种测试方法进行说明。

2.2.1 边界值

任何的程序都可以看作是一个函数，程序的输入构成了函数的定义域，而程序的输出构成了函数的值域。将定义域按照取值范围进行划分区间，进而选择输入项，就形成了边界值测试的雏形。边界值分析通常关注的是输入空间的边界，同时实践也证明了缺陷更可能出现在输入变量的极值附近出现。

边界值分析的基本思想是使用最小值、略高于最小值、正常值、略低于最大值、最大值为输入变量，通常记为（min、min+、nom、max-、max），在考虑测试的健壮性的情况下，增加略大于最大值（max+）和略小于最小值（min-）的取值。

单缺陷假设是边界值分析的关键假设。单缺陷假设指“失效极少是由两个或两个以上的缺陷同时发生引起的”。在边界值分析中，单缺陷假设即选取测试用例时仅仅使得一个变量取极值，其他变量均取正常值；

多缺陷假设，则是指“失效通常由两个或两个以上缺陷同时作用引起的”，要求在选取测试用例时同时让多个变量取极值。

	单缺陷假设	**多缺陷假设**
正常值	基本边界值测试	最坏情况测试
健壮值	健壮性测试	健壮最坏情况测试

下面对于边界值分析进行一个简单的说明，其中：

a<=X1<=b

c<=X2<=d

- 基本边界值测试

基本边界值测试，在设计测试用例时，使一个变量在数据有效区内取 min、min+、nom、max-、max。

基于单缺陷假设和正常值，对于有 n 个输入变量的程序，基本边界值分析的测试用例个数为 4n+1。

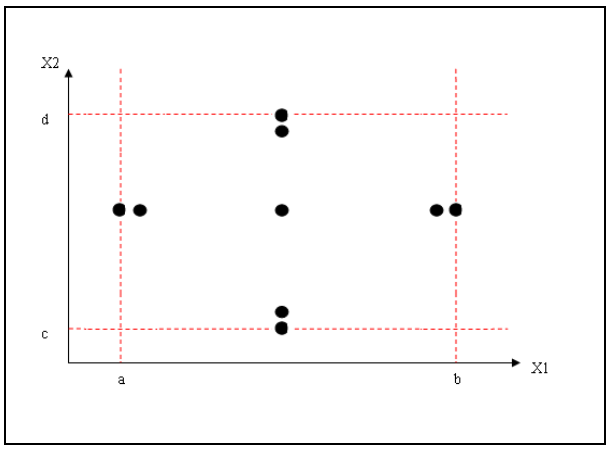

图 2-1：基本边界值

- 健壮性测试

健壮性是指在异常情况下，软件还能正常运行的能力。健壮性测试是基本边界值的一种简单扩展，除了变量的 5 个边界分析取值还要考虑 max+和 min-。健壮性测试的最大价值在于异常情况的处理，它是检测软件系统容错性的重要手段。

基于单缺陷假设和健壮值，对于有 n 个输入变量的程序，健壮性测试的测试用例个数为 6n+1。

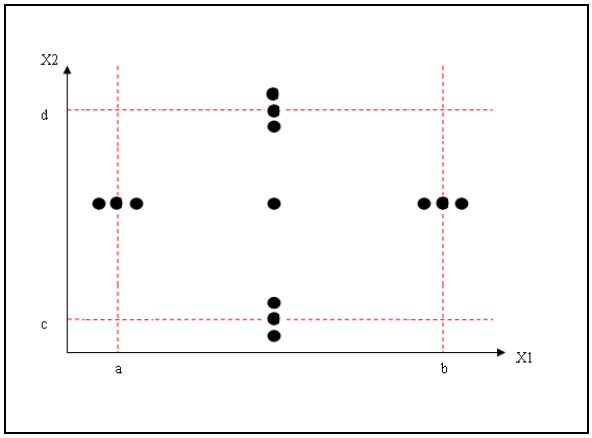

图 2-2：健壮性边界值

- 最坏情况测试

最坏情况测试关心的是当多个变量取极值时出现的情况。最坏情况测试中，对每一个输入变量首先进行 min、min+、nom、max-、max 这 5 个元素集合的测试，然后对这些集合进行笛卡尔积计算，以生成测试用例。

基于多缺陷假设和正常值，对于有 n 个输入变量的程序，最坏情况测试的测试用例个数为 5^n。

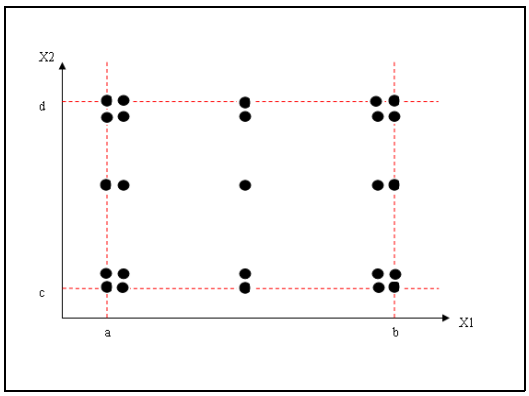

图 2-3：最坏情况边界值

- 健壮最坏情况测试

健壮最坏情况假设对每一个变量首先进行 min-、min、min+、nom、max-、max、max+这 7 个元素的集合。然后对这些集合进行笛卡尔积运算，以生成测试用例。

基于多缺陷假设和健壮值，对于有 n 个输入变量的程序，健壮最坏情况测试的测试用例个数为 7^n。

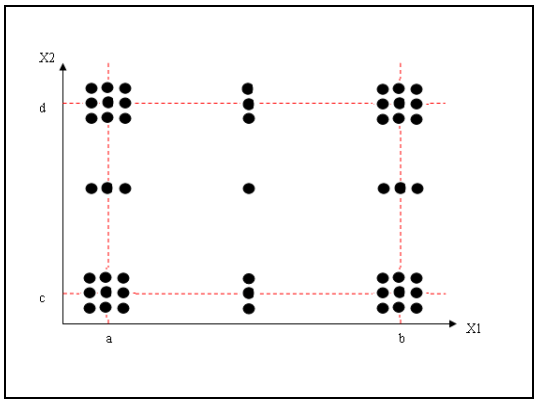

图 2-4：健壮最坏情况边界值

边界值的方法通常针对一些输入域的空间，一般情况下是一个实数的范围，但在项目中系统的状态、界面的状态、数据库的状态，这些都有一些边界值，我们同样可以应用边界值的方法来考虑。比如某个输入界面下，重置按钮的功能测试，此时我们可以将刚刚进入此界面没有任何操作、输入-〉选择重置后作为界面的边界值，这些情况下的重置功能更容易出现错误。

2.2.2 等价类

等价类测试源于集合的划分，划分是指互不相交的一组子集，而这些子集的并是整个集合。这对于测试有非常重要的意义，覆盖整个集合提供了一种完备性，互不相交保证了一种无冗余性。

等价类是指某个输入域的子集合，在该子集合中，各个输入数据对于揭露程序中的错误都是等效的。等价类测试用例编写的步骤如下：

- 先考虑数据的数据类型（合法类型和非法类型）
- 再考虑数据范围（合法类型中的合法区间和非法区间）
- 区分等价类
- 从每一个等价类中选举一个测试数据构造测试用例

等价类按照健壮性和单/多缺陷假设分为弱一般等价类、强一般等价类、弱健壮等价类、强健壮等价类，如下表：

	单缺陷假设	**多缺陷假设**
有效性	弱一般等价类	强一般等价类
健壮性	弱健壮等价类	强健壮等价类

下面对于等价类进行一个简单的说明，其中：

a<=X1<=d，区间为[a，b），[b，c），[c，d]

e<=X2<=g，区间为[e，f），[f，g]

- 弱一般等价类测试

弱一般等价类测试中，基于变量的有效区间和单缺陷假设，保证每一个变量都覆盖了各自的有效区间，即（1）覆盖每个变量的有效等价类；（2）不考虑变量等价类的组合。

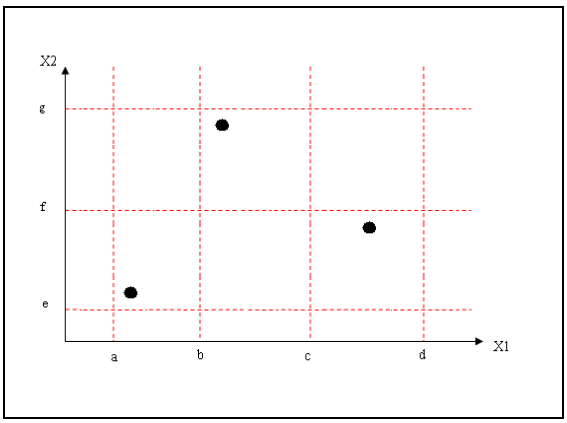

图 2-5：弱一般等价类

- 强一般等价类测试

强一般等价类测试中，基于变量的有效区间和多缺陷假设，覆盖各个变量有效等价类的所有组合。即（1）覆盖每个变量的有效等价类；（2）考虑变量等价类的组合。

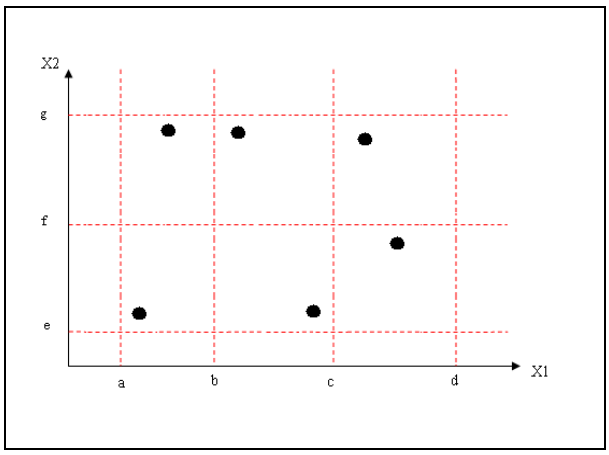

图 2-6：强一般等价类

- 弱健壮等价类测试

弱健壮等价类测试中，基于变量的有效/无效区间和单缺陷假设，覆盖各个变量有效/无效等价类。即（1）覆盖每个变量的有效/无效等价类；（2）不考虑变量等价类的组合；（3）对于无效输入，用例拥有一个无效值，并保持其余的值都是有效的。

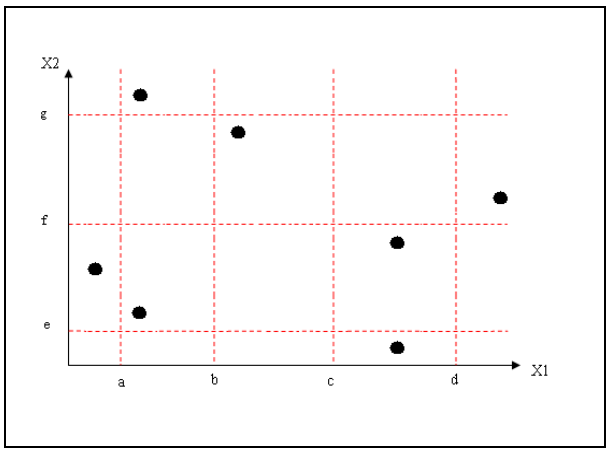

图 2-7 弱健壮等价类

- 强健壮等价类测试

强健壮等价类测试中，基于变量的有效/无效区间和多缺陷假设，需要覆盖各个变量有效/无效等价类。即（1）覆盖每个变量的有效/无效等价类；（2）考虑变量等价类的组合。

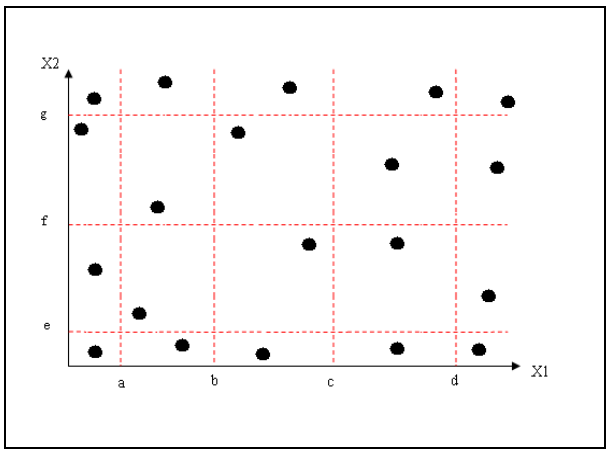

图 2-8 强健壮等价类

理论上来说，如果等价类里面的一个数值能够发现缺陷，那么该等价类里面的其他数值也能够发现该缺陷。

在项目中常用的的等价类划分方法：

- 如果规定了输入值的范围(闭区间)，可以分为一个有效等价类，两个无效的等价类。
- 如果输入是布尔表达式，可以分为一个有效等价类和一个无效等价类。
- 如果规定了输入数据的一组值，而且程序对不同输入值做不同的处理，则每个允许的输入值是一个有效的等价类，此外还有一个无效的等价类（任意一个不允许的输入值）。
- 如果规定了输入数据必须遵循的规则，可以划分出一个有效的等价类（符合规则）和若干个无效的等价类（从不同角度违反规则）。

2.2.3 决策表

基于决策表的测试方法是严格的，决策表具有逻辑严格性。决策表测试适合具有以下特征的应用程序：

- if-then-else 逻辑很突出
- 输入变量之间存在逻辑关系
- 涉及输入变量子集的计算
- 输入和输出之间存在因果关系

通常我们把条件解释为输入，把行动解释为输出。决策表有四个部分，形式如下：

桩	规则 1	规则 2	规则 3、4	规则 5	规则 6	规则 7、8
C1	T	T	T	F	F	F
C2	T	T	F	T	T	F
C3	T	F	-	T	F	-

A1	X	X		X		
A2	X				X	
A3		X		X		
A4			X			X

表格中黄色背景部分为条件桩，紫色背景部分为条件项，绿色背景部分为行动桩，茶色背景部分为行动项。

条件桩（Condition Stub）：列出了问题的所有条件，通常认为列出的条件的次序无关紧要。

动作桩（Action Stub）：列出了问题规定可能采取的操作，这些操作的排列顺序没有约束。

条件项（Condition Entry）：列出针对条件桩对应条件的取值，在所有可能情况下的真假值。

动作项（Action Entry）：列出在条件项的各种取值情况下应该采取的动作。

判定表的建立步骤：

- 确定输入输出的等价类，整理出条件桩和动作桩
- 填入条件项
- 填入动作项，形成初始判定表
- 简化，合并相似规则

2.2.4 小结

介绍了三种常用的测试方法，在项目中我们需要考虑如何有效地编写用例，哪种方法更适合，如何利用各种方法的优势相互补充整理出最佳的测试方案。在测试设计时需要根据业务模型、状态迁移、验证条件等多方面考虑，选择合适的测试方法，通常情况下，测试方法选择的原则是：

- 如果变量引用的是物理量，可采用定义域测试和等价类测试。
- 如果变量是独立的，则可以用定义域测试和等价类测试。
- 如果变量不是独立的，可采用决策表测试。
- 如果可以保证是单缺陷假设，可采用边界值分析和健壮性测试。
- 如果可以保证是多缺陷假设，可采用最坏情况测试、健壮最坏情况测试和决策表测试。
- 如果程序包含大量例外处理，可采用健壮性测试和决策表测试。

- 如果变量引用的是逻辑量，可采用等价类测试和决策表测试。

对于测试用例的执行顺序，同样可以从测试用例设计方法的角度来考虑，比如对边界值法，基于单缺陷假设、正常值的情况的用例主要验证系统能够工作，这部分用例可以优先测试，确认系统能够实现基本功能。基于多缺陷假设和健壮性的用例主要确认系统的健壮性，这部分用例可以在确保基本功能正确的前提下进行测试。

3.附录

参考资料

《The Art of Software Testing》--Grenford J. Myers

《Testing Computer Software,Second Edition》 --Cem Kaner,Jack Falk,Hung QuocNguyen

《Software Testing A Craftsman's Approach(Second Edition)》-- Paul C.Jorgensen

《Software Testing》 --Ron Patton

《Effective Methods for Software Testing,Second Edition》-- William E.Perry

《Surviving the Top Ten Challenges of Software Testing:A People-Oriented Approach》 -- William E.Perry, Randall W.Rice

《Managing the Testing Process(Second Edition)》-- Rex Black

《测试用例设计白皮书》-- Vince