

软件功能测试用例设计方法的探讨

杨 彬

(辽宁行政学院信息技术系 辽宁沈阳 110161)

摘要: 阐述了采用软件功能测试用例的必要性, 然后通过使用现有的软件功能测试用例的具体设计方法(等价分类法、边界值分析法、因果图法以及错误推测法)对具体实例的分析, 提出了软件功能测试的相关技巧。

关键词: 功能测试; 测试用例; 设计。

中图分类号: TP311 文献标识码: A 文章编号: 1673- 4629(2007)06- 0023- 04

软件开发项目在经历了计划、需求分析、设计和编码以后, 已经取得了阶段性的成果, 然而, 这些阶段性成果能否满足客户的需求以及能在多大程度上满足客户的需求仍然是个未知数^[1]。这时就必须进行包括功能测试在内的大量的软件测试。

功能测试是指在对程序进行的功能抽象的基础上, 将程序划分成功能单元, 然后在数据抽象的基础上, 对每个功能单元生成测试数据进行测试^[2]。用这种方法进行测试时, 被测程序被当作打不开的黑盒, 因而无法了解其内部构造, 因此又称为黑盒测试。设计测试用例要针对特定功能或组合功能的测试方案, 并编写成文档, 测试用例的选择既要有一般情况, 也应有极限情况以及最大和最小的边界值情况。在测试时, 完全不考虑程序内部结构和内部特性, 测试者在程序接口进行测试, 它只检查程序功能是否按照需求规格说明书的规定正常使用, 以及检测程序是否能适当地接收输入数据而产生正确的输出信息, 并且保持外部信息(如数据库或文件)的完整性^[3]。

1 功能测试用例设计方法

在进行功能测试时, 必须站在最终用户的立场上, 检验输入输出信息及系统性能指标是否符合规格说明书中有关功能需求的规定。而具体测试用例的设计则必须依据项目资料中的功能描述, 把整个系统分解出若干小的“功能点”, 并映射出测试用例。由于系统“功能点”往往非常繁多, 组合起来更是难以计数, 最终将导致庞大的测试用例集。因此有必要采取一些有效的功能测试用例设计的方法和技巧, 以减少测试用例规模, 提高测试效率。本文将探讨常用的功能测试用例的设计方法, 包括: 等价分类法、边界值分析法、错误推测法和因果图法。

1.1 等价分类法

所谓等价分类法, 就是把输入数据的可能值划分为若干等价类, 使每一类中任何一个测试用例, 都能代表同一

等价类中的其它测试用例^[4]。这样就把漫无边际的随机测试变成有针对性的等价测试了。

(1) 设计测试用例的原则:

为每一个等价类规定一个唯一的编号。

设计一个新的测试用例, 使其尽可能多地覆盖尚未被覆盖的有效等价类, 重复这一步。直到所有的有效等价类都被覆盖为止。

设计一个新的测试用例, 使其仅覆盖一个尚未被覆盖的无效等价类, 重复这一步。直到所有的无效等价类都被覆盖为止。

(2) 等价类划分举例:

某保险公司承担人寿保险已有多年历史, 该公司保费计算方式为: 投保额×保险率, 保险率又依点数不同而有别, 10点以上费率为0.6%, 10点以下费率为0.1%。保险率点数分配情况见表1。

表 1 保险率点数分配表

年龄	20~39岁	6点
	40~59岁	4点
	60岁以上或20岁以下	2点
性别	Male	5点
	Female	3点
婚姻	已婚	3点
	未婚	5点
抚养人数	1人扣0.5点最多扣3点(四舍五入取整)	

第一步: 根据输入数据, 划分等价类, 划分时不仅要考虑代表“有效”输入值的有益等价类, 还需考虑代表“无效”输入值的无益等价类。划分结果见表2。

第二步: 根据分析结果设计测试用例, 每一个无益等价类至少要用一个测试用例, 不然就可能漏掉某一类错误, 但允许若干有益等价类合用同一个测试用例, 以便进一步减少测试的次数。测试用例见表3。

收稿日期: 2007—06—30

作者简介: 杨彬(1975—), 女, 辽宁辽阳人, 讲师, 研究方向: 计算机应用与软件工程。

表 2 等价分类表

数据	有益等价类	无益等价类
1 年龄	20 -39 任选一个	
2 年龄	40 -59 任选一个	
3 年龄	60 岁以上或 20 岁以下任选一个	小于 1 选一个,大于 99 选一个
4 性别	英文 Male、M、F、Female 任选一个	非英文字如 男」
5 性别	英文 Male、M 任选一个	非 Male、M、Female、F 之任意字元,如 Child」
6 性别	英文 Female、F 任选一个	非 Male、M、Female、F 之任意字元,如 Child」
7 婚姻	已婚」	非 已婚」或 未婚」之任意字符,如 离婚」
8 婚姻	未婚」	非 已婚」或 未婚」之任意字符,如 离婚」
9 扶养人数	空白	
10 扶养人数	1-6	小于 1 选一个
11 扶养人数	7-9	大于 9 选一个
12 保险费率	10 点以上(0.6%)	
13 保险费率	10 点以下(0.6%)	

表 3 测试用例

用例编号	年龄	性别	婚姻	扶养人数	保险费率	备注
1	27	Female	未婚	空白	0.6%	有益等价类
2	50	Male	已婚	2	0.6%	有益等价类
3	70	F	未婚	7	0.1%	有益等价类
4	0	M	已婚	4	无法推算	年龄类无效,因此无法推算保险费率
5	100	Female	未婚	5	无法推算	年龄类无效,因此无法推算保险费率
6	1	男	已婚	6	无法推算	年龄类无效,因此无法推算保险费率
7	99	Child	未婚	1	无法推算	性别类无效,因此无法推算保险费率
8	30	Male	离婚	3	无法推算	婚姻类无效,因此无法推算保险费率
9	75	Female	未婚	0	无法推算	扶养人数无效,因此无法推算保险费率
10	17	Male	已婚	10	无法推算	扶养人数无效,因此无法推算保险费率

1.2 边界值分析

边界值分析是一种补充等价划分的测试用例设计技术,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。实践证明,在设计测试用例时,对边界附近的处理必须给予足够的重视,为检验边界附近的处理需专门设计测试用例,这样在测试时可以取得良好的效果。

(1) 设计测试用例的原则:

如果输入条件规定了取值范围,应以该范围的边界及略大于和略小于边界的值作为测试用例。例如,输入值范围是[1,2],应测试 0.99,1,1.01,1.99,2,2.01 这几个输入数据的执行情况。

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

针对每个输出条件使用上述 2 条原则。

如果程序规格说明中提到的输入输出域是个有序的集合(如顺序文件、表格等),就应注意选取有序集的第一个和最后一个元素作为测试用例。

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。

分析规格说明,找出其他的可能边界条件。

(2) 边界值分析举例

在三角形问题描述中,除了要求边长是整数外,没有给出其它的限制条件。在此,我们将三角形每边边长的取值范围设定为[1, 100],设计的测试用例见表 4。

表 4 边界值分析测试用例

用例编号	边长 a	边长 b	边长 c	预期输出
1	60	60	1	等腰三角形
2	60	60	2	等腰三角形
3	60	60	60	等边三角形
4	50	50	99	等腰三角形
5	50	50	100	非三角形
6	60	1	60	等腰三角形
7	60	2	60	等腰三角形
8	50	99	50	等腰三角形
9	50	100	50	非三角形
10	1	60	60	等腰三角形
11	2	60	60	等腰三角形
12	99	50	50	等腰三角形
13	100	50	50	非三角形

1.3 因果图法

当被测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组合时, 用因果图可直观的表明输入条件和输出动作之间的因果关系, 能帮助测试人员把注意力集中到与程序有关的那些输入组合, 比采用等价分类法有更高的测试效率。

(1) 设计测试用例的原则

由于语法或环境限制, 有些原因与原因之间, 原因与结果之间的组合情况不可能出现。为表明这些特殊情况, 在因果图上用一些记号表明约束或限制条件。

把因果图转换为判定表。

把判定表的每一列拿出来作为依据, 设计测试用例。

(2) 因果图法举例^[4]

某规格说明规定: 输入的第一列必须是字母 A 或 B, 第二列输入必须是一个数字, 第一、二列满足条件执行操作 H; 如果第一列不正确, 则给出信息 L; 如果第二列不正确, 则给出信息 R。

第一步: 分析输入/输出情况, 并给每一种情况编号, 见表 5。

表 5 输入/输出情况

输入情况	输出结果
1: 第一列是字母 A	21: 执行操作 H
2: 第一列是字母 B	22: 给出信息 L
3: 第二列是一数字	23: 给出信息 R

由于原因 1 和原因 2 不能同时成立, 可用 E 约束; 如果第一列符合条件, 还应检查第二列是否正确。这里将第一列正确的情况编号设为 11, 作为中间结果。

第二步: 画出因果图, 见图 1。

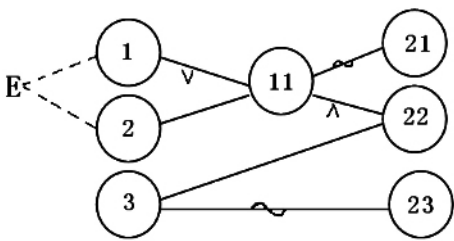


图 1 因果图

第三步: 根据因果图建立判定表, 如表 6 所示。其中表右半部中间用 1 表示条件成立, 0 表示条件不成立。表的下方列出针对可能出现的六种情况而设计的测试用例。

1.4 错误推测法

错误推测法在很大程度上是凭经验进行的, 是凭人们对过去所做的测试工作结果的分析, 对所揭示的缺陷的规律性做直觉的推测来发现缺陷的^[9]。

(1) 设计测试用例的原则

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, 根据它们选择测试用例。例如, 输入数据为 0 是容易发生错误的情形, 因此可选择输入数据为 0 作为测试用例。

表 6 判定表

组合条件	1	2	3	4	5	6	7	8
输入情况 1	1	1	1	1	0	0	0	0
2	1	1	0	0	1	1	0	0
3	1	0	1	0	1	0	1	0
11			1	1	1	1	0	0
输出结果 22			0	0	0	0	1	0
21			1	0	1	0	0	0
23			0	1	0	1	0	1
测试用例			A8	A?	B4	BN	C2	DA
			A3	AM	B7	B!	X6	P;

也可以针对一个排序程序, 如: 输入空的值(没有数据)、让所有的输入数据都相等、让所有输入数据有序排列、让所有输入数据逆序排列等。

(2) 错误推测法举例

测试一个对线性表(比如数组)进行排序的程序, 可推测列出以下几项需要特别测试的情况:

- I. 输入的线性表为空表;
- II. 表中只含有一个元素;
- III. 输入表中所有元素已排好序;
- IV. 输入表已按逆序排好;
- V. 输入表中部分或全部元素相同。

2 功能测试的技巧

在初次接触系统时最好先不看功能说明, 而是凭借自己的经验以及有限的帮助来进行操作, 然后再按照功能说明一步一步测试, 将系统中的实现方式等内容与自己判断的进行比较, 这样有助于发现更多的错误^[6]。

(1) 在开始测试时应保证数据的正确性。

(2) 输入不合法的数据, 观察系统是否能正确处理异常数据。

(3) 代码重用测试, 在开发过程中有些模块功能几乎相同, 程序员在重用代码时可能忘记在原有代码上修改或修改不全面, 而造成的错误。

(4) 外界环境变化测试。有些系统在开发时依赖于另外一个系统, 当在一个新的系统平台上运行, 可能会发生错误。

(5) 在程序员刚修复 Bug 之后的地方, 再找一找, 往往程序员只修复报告出来的缺陷而不去考虑别的功能在修改时可能会重新造成错误。

(6) 认真做好测试记录。在做完一天的测试记录之后, 第二天再根据第一天的测试记录重复测试也许你会发现有未修正的错误。

3 结论

软件测试是通过测试用例的运行, 尽可能多地发现并

纠正错误, 以确保最终交付的产品符合用户的需求。而在大多数情况下, 要想对所有的用例进行彻底的测试是非常困难的, 所以有必要通过应用上述的方法设计出典型的测试用例并进行有针对性的测试, 从而达到既提高测试效率又能保证产品质量的目的。

参考文献:

[1]Binder R V. Testing Object- Oriented Systems [M]. Massachusetts: Addison- Wesley, 2000.
[2]史济民, 顾春华, 李昌武, 苑荣. 软件工程—原理、方

法与应用[M]. 2 版. 北京: 高等教育出版社, 2002.

[3]肖冬娟, 赫克刚, 葛玮, 等. 软件测试中的测试用例及复用研究[J]. 计算机技术与发展, 2006, 16(1): 69- 72.

[4]陆惠恩, 陆培恩. 软件工程[M]. 2 版. 北京: 电子工业出版社, 2002.

[5]韩丽娜. 黑盒测试及测试工具 Rational Robot 的应用[J]. 计算机工程与设计, 2006, 27(2): 359- 360.

[6]林钦, 潘健鸿. 对软件测试的进一步研究及探索[J]. 中国科技信息, 2005(16): 40.

责任编辑: 邱维敦]

Study on Design Method of Software Function Test Cases

YANG Bin

Abstract: This paper first outlines the necessity of using software function test cases, and then puts forward some relative techniques of software function test based on the practice of using some present design methods (equivalence partitioning, boundary value analysis, cause- effect map and error guessing method) on several software function test cases.

Key words: function test; test cases; design