

## 基于需求特征的软件测试用例设计模型研究<sup>①</sup>

李文武<sup>②\*</sup> 李文法<sup>③\*\*</sup> 段冰毅<sup>\*</sup>

( \* 北京航空航天大学计算机学院 北京 100191 )

( \*\* 北京联合大学信息学院 北京 100101 )

**摘 要** 针对软件测试设计过程中,不同测试人员对同一类特征的测试需求进行测试用例设计时测试设计质量和有效性问题,提出了一种基于需求特征的测试用例设计模型 *TsetCaseDesignModel*。该模型为  $\langle ModelIndex, ModelInfo, TestDataBody, TestProBody, TestRunbody \rangle$  五元组, *ModelIndex* 为测试用例设计模型代表的需求特征和测试类型, *ModelInfo* 描述测试用例设计模型的基本信息, *TestDataBody* 为模型数据体, *TestProBody* 是模型的测试规程体, *TestRunbody* 是模型的测试执行体。该模型集成了针对同类测试需求进行测试用例设计时的全部设计要素,具有动态性、知识性、规范性和易维护四个特点。对基于该模型和基于人工方式开展测试用例设计的有效性和效率的评价表明,该模型能有效提高测试用例设计质量和效率。

**关键词** 需求特征,测试用例设计模型,开发,评价

## 0 引言

软件测试设计是软件测试的一个关键过程,它以测试项为输入,按照软件测试设计流程,利用各种软件测试设计方法,设计出高质量的测试用例产品。测试人员在进行测试设计时,有三个方面需要特别关注:一个是测试设计结果的规范性,即测试用例结构及组成必须满足测试技术要求;二是测试设计结果的有效性,即测试用例的内容能够达到满足测试需求,发现软件缺陷的目的;三是测试设计效率,即测试设计工作需要适应测试周期紧张的情况。如何既保证测试设计结果的规范性和有效性,又能同时保证测试效率,做到测试设计质量和测试设计效率相统一,是每个专业软件测评机构和测试人员面临的重大问题。

要满足上述三方面的要求并不容易。在实际测试设计过程中,需要测试设计人员具有丰富的测试设计经验和高超的测试设计技巧,即使有辅助测试用例设计工具帮助,使测试设计的规范性得到提高,

测试设计结果有效性和效率最终还要靠测试人员的经验和能力决定。测试设计的有效性需要解决测试设计质量稳定性和测试需求多样性两个问题。对于测试设计质量稳定性,不同测试人员因测试设计能力差异会导致对同一类测试需求的测试设计质量稳定存在偏差。对于测试需求多样性,传统的等价类划分、因果图、边界值等测试设计方法已不能满足测试设计需求,不同领域、不同业务方向、不同技术体制的测试需求要求测试设计人员能设计出紧贴测试需求的个性化测试用例产品。解决测试设计质量稳定性和测试需求多样性所带来的测试设计有效性问题,需要研究探索新的技术手段和方法。综合国内外研究成果,主要有两种技术途径。一种是采用测试用例自动生成技术,即基于规格说明或测试需求自动生成软件测试用例。文献[1]提出了一种用关系代数查询表示的规格说明中自动生成测试用例的方法,文献[2]提出了一种基于布尔规格说明的测试用例自动生成方法,文献[3]提出了一种基于 Z 规格说明的软件测试用例自动生成方法,文献[4]提出了一种基于 UML 的软件测试用例自动生成方

① 国家 863 计划(2011AA040506)和北京联合大学新起点计划(ZK201204)资助项目。

② 男,1975 年生,博士;研究方向:软件测试。

③ 通讯作者, E-mail: liwenfa@ buu. edu. cn  
(收稿日期:2013-02-10)

法。测试用例自动生成方法能自动完成测试设计工作,不需要测试人员具有丰富的测试经验和较高的专业水平,这样可较好地解决测试设计人员技术能力差异性带来的测试设计质量稳定性问题。该方法不足之处在于需要利用形式化的方法表示测试需求,在工程实践上有局限性。第二种技术途径是测试用例复用。文献[5]从面向复用的角度介绍了软件资产与过程管理的有关内容,纯拷贝式的测试用例复用能在一定程度上解决测试设计人员技术能力差异性和测试需求多样性问题,该方法不足之处在于缺乏灵活性,复用层次较低,应用范围小,对已有测试用例产品核心设计精华提炼不够,在回归测试中该方法应用广泛<sup>[6]</sup>,但对于新软件测试很难推广应用。针对上述两种技术的有效性和效率的不足,本文提出了一种基于需求特征的测试用例设计模型 *TsetCaseDesignModel*,该模型解决了两个问题:(1)为同类测试需求提供了测试设计详细指南和参考;(2)通过模型的测试规程体和执行体来适应同类测试需求的差异性,为自动生成测试用例提供了方法基础,解决了测试设计质量稳定性和效率问题。因此,*TsetCaseDesignModel* 对于提高测试用例设计的有效性和效率有重要意义。

## 1 测试用例设计模型

### 1.1 模型结构

测试用例设计模型的组成结构如图 1 所示。测试设计模型包括模型索引、模型基本信息、数据体、测试规程体和测试执行体 5 个部分。模型索引为测试用例设计模型与测试项的对接提供匹配信息,模型基本信息提供测试用例的初始化要求、前提约束、用例期望和终止要求内容,数据体包含了测试用例设计模型处理的测试数据(字符串、数值、时间等),

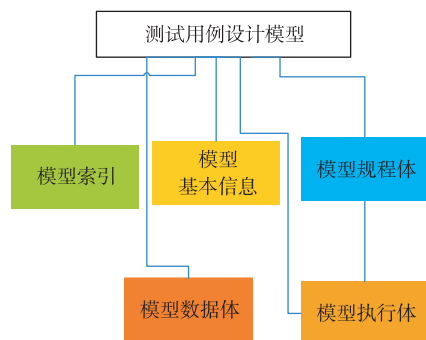


图 1 测试用例设计模型组成结构图

测试规程体给出测试用例设计模型所代表的测试规程即测试步骤,测试执行体包含了根据数据体和测试规程体产生测试执行实例的方法。

### 1.2 模型表示

采用形式化方法表示测试用例设计模型,假设 *TsetCaseDesignModel* 为测试用例设计模型,则 *TsetCaseModel* 可以表示为如下所示的 5 元组  $\langle ModelIndex, ModelInfo, TestBody, TestProBody, TestRunbody \rangle$  每个组成要素描述如下:

(1) *ModelIndex* 为模型代表的需求特征和测试类型, *ModelIndex* 在测试用例设计模型的应用过程中起到关键作用,是测试用例设计模型具备可用性的前提和基础。*ModelIndex* 中的信息与测试项中需求特征和测试类型信息保持一致,为测试用例设计模型的选取提供唯一索引。*ModelIndex* 表示为一个二元关系,  $ModelIndex = [ReqTz \times TestType]$ ,其中, *ReqTz* 是需求特征集合, *TestType* 是 23 种类型的集合, *ModelIndex* 中的每个元素可以作为 *TsetCaseDesignModel* 的模型索引使用。

(2) *ModelInfo* 是测试用例设计模型的基本信息,可以表示为一个 3 元组  $\langle ModelObjective, ModelLim, ModelExpec \rangle$ ,其中:

- 1) *ModelObjective* 表示测试用例设计模型的目的;
- 2) *ModelLim* 表示模型的前提和约束;
- 3) *ModelExpec* 表示为模型期望,它给出测试设计模型使用后期望达到的结果。

(3) *TestBody* 为模型数据体,它是测试用例设计模型处理的测试数据(如字符串、数值、时间等)集合,用下式表示:

$$TestBody = \left\{ \begin{array}{l} Text(X1, Y1, LogicExp1) \\ Number(X2, Y2, LogicExp2) \\ \vdots \\ dataN(XN, YN, LogicExpN) \end{array} \right\} \quad (1)$$

这里 *dataN* 表示在测试用例设计时测试数据需要考虑的具体测试数据集合, *XN*、*YN* 是特征数据的边界范围, *LogicExpressionN* 是测试数据约束条件。不同类别的测试数据包含不同的测试数据值,如字符串类型 *dataN(XN, YN, LogicExpN)* 的测试数据应包括正常数据、非法数据、边界数据等<sup>[7]</sup>,可以根据需要进行定制和扩充。

(4) 测试规程体 *TestProBody* 是测试用例设计模型的核心部件之一,是针对特定测试需求特征的

测试用例设计方法、步骤、经验、知识的提炼和总结。 $TestProBody = TestStepProduct(InputDataSet, TestStep)$ , 此等式的含义是,  $TestProBody$  代表测试执行步骤产生的过程。该过程类似高级程序设计语言的过程, 该过程有两个参数  $InputDataSet$  和  $TestStep$ ,  $InputDataSet$  为形参,  $TestStep$  为实参。 $InputDataSet$  为从测试项中提取的输入数据集合,  $InputDataSet = \{DataItem1, DataItem2 \cdots DataItemN\}$ ,  $DataItemN$  是测试项处理的一个数据项, 该过程实现的功能是根据  $InputDataSet$ , 动态生成测试规程及测试执行步骤。该过程的输出为测试执行步骤, 表示为  $TestStep$ ,  $TestStep = [action \times object \times data]$ , 其中,  $action$  是操作动作集合, 典型的操作包括鼠标单击、鼠标双击、输入等,  $object$  是操作对象集合, 典型的对象包括按钮、菜单、文档框、列表、单选按钮等,  $data$  是对象属性值的集合, 从  $TestStep$  定义可以看出,  $TestStep$  是三元关系<sup>[8]</sup>, 例如:

$$TestStep = \begin{Bmatrix} click, button, OK \\ click, editbox, name \\ \vdots \\ input, name, beijing \end{Bmatrix} \quad (2)$$

将测试用例设计模型中的  $TestProBody$  表示为一个过程体, 体现出测试用例设计模型对特征相同但输入数据项不同的测试项的适应性。对于相同的需求特征, 不同软件需求项处理的数据类型和数量存在差异, 如何在测试用例设计阶段通过测试用例设计模型来屏蔽这些差异,  $TestProBody$  是重要技术手段。利用  $TestProBody$  抓住该需求特征关键的测试用例设计方法, 用程序设计语言中的过程来动态完成测试规程的实现, 为测试用例设计模型在其需求特征范围内的推广应用奠定了坚实的技术和结构基础。同程序设计语言中过程包含的结构类似,  $TestProBody$  可包含顺序、分支和循环三种结构。

(5) 测试执行体  $TestRunbody$  是根据数据体和测试规程体产生测试执行实例的实现过程。它同样是测试用例设计模型的核心部件, 在测试规程体确定的条件下, 根据测试项所处理数据项和数据体内容, 选择典型的测试数据生成测试执行实例。 $TestRunbody$  是后续阶段测试执行开展依据。 $TestRunbody = TestInstanceProduct(InputDataSet, TestInstance)$ , 上面等式的含义是,  $TestRunbody$  是一个测试执行实例产生的过程, 该过程同高级程序设计语言的过程概念相同。该过程有两个参数, 分别为  $InputDataSet$  和  $TestInstance$ ,  $InputDataSet$  为

形参,  $TestInstance$  为实参。 $InputDataSet$  是从测试项中提取的输入数据集合,  $InputDataSet = \{DataItem1, DataItem2 \cdots DataItemN\}$ 。 $DataItemN$  是测试项处理的一个数据项, 该过程实现的功能是以  $InputDataSet$  为输入, 在各个  $DataItem$  的测试数据集中选择典型测试数据动态生成测试执行实例。该过程的输出为测试执行实例集合, 表示为  $TestInstance$ ,  $TestInstance = \{InsData1, InsData2 \cdots InsDataN\}$ ,  $InsDataN$  是一个测试执行实例包含的测试数据集合。 $TestInstance$  与  $InputDataSet$  的关系如图 2 所示。

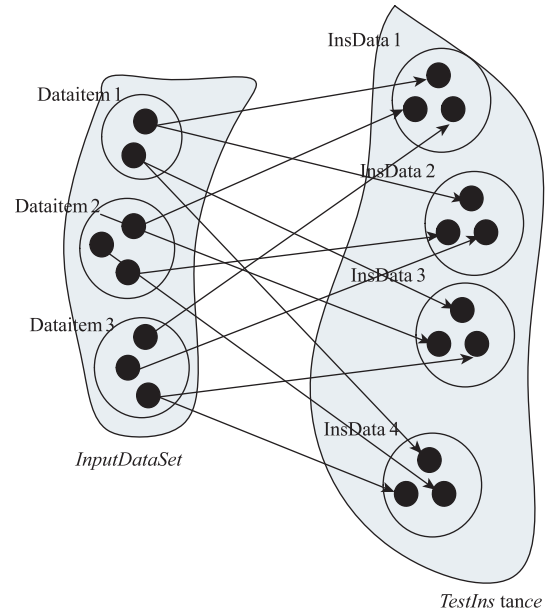


图2 测试项数据与测试执行实例映射关系

## 2 测试用例设计模型实例

下面通过数据录入功能测试用例设计模型给出一个测试用例设计模型的实例。数据录入功能在软件功能设置上比较普遍, 数据录入功能的需求特征为用户按要求录入各种数据后保存入库, 即用户在软件提供的窗口中输入各种数据后, 软件将数据保存入库, 之后用数据浏览、后台查询等方式进行验证。数据录入功能通常的测试类型为功能测试, 数据录入测试用例的测试步骤特征包括进入数据添加界面, 逐个输入数据项, 点击保存, 浏览验证等。针对该需求特征的测试用例设计模型  $TsetCaseDesign-Model$  表示如下:

(1)  $ModelIndex$  为该测试用例设计模型代表的需求特征和测试类型,  $ModelIndex = [addrecord, gnacs]$ 。 $addrecord$  为需求特征,  $gnacs$  为功能测试。

(2) *ModelInfo* 是测试用例设计模型的基本信息, 可以表示为一个 3 元组,  $\langle ModelObjective, ModelLim, ModelExpec \rangle$ , 其中:

1) *ModelObjective* 表示测试软件添加数据功能是否正常。

2) *ModelLim* 表示模型的前提和约束, *ModelLim* = 软件启动, 并进入数据添加窗口。

3) *ModelExpec* 表示为模型期望, *ModelExpec* = 软件可以实现合法数据的添加, 对非法数据实施有效处理和提示。

(3) *TestDataBody* 为模型数据体。模型数据体是测试用例设计模型处理的特征数据(如字符串、数值、时间等)集合, 其表示如下:

$$TestDataBody = \left\{ \begin{array}{l} Text(X1, Y1, LogicExp1) \\ Number(X2, Y2, LogicExp2) \\ \vdots \\ List(XN, YN, LogicExpN) \end{array} \right\} \quad (3)$$

(4) 采用类 pascal 语言描述 *TestStepProduct* 过程如下:

```
TestStepProduct ( InputDataSet, TestStep )
{
  begin
    // 设置计数器 Count, 对 InputDataSet 中的元素个数进行计数
    Count = 1
    // TestStep 初始化
    TestStep = [ open, windows, adddata ];
    计算 InputDataSet 的大小, 计为 long;
    While long ≥ 1
    {
      // 返回 InputDataSet 中的数据项
      DataItem = InputDataSet ( Count )
      // TestStep 增加一个步骤, TestStep = TestStep +
      < input, DataItem( name ), DataItem( data ) >
      long = long - 1
      Count = Count + 1
    }
  }
}
```

(5) 测试执行体 *TestRunbody* = *TestInstanceProduct* ( *InputDataSet*, *TestInstance* )。 *InputDataSet* 含  $K$  个测试数据项:  $A1, A2, \dots, Ak$ , 其中的每个测试数据项又分别生成了  $N1, N2, \dots, Nk$  个测试数据值。则全部执行方案有  $N1 \times N2 \times \dots \times Nk$  个, 测试执行方案数量巨大<sup>[9]</sup>, 有一些执行方案的测试有效性相同, 存在大量冗余。

*TestInstanceProduct* 过程的目的是实现测试执行实例自动生成算法, 生成具有意义的最小执行实例子集。 *TestInstanceProduct* 每一次只关注一个测试数据项的测试, 包括测试它的有效数据、无效性数据、边界数据以及敏感数据, 其余数据均取有效值。如图 3 所示, *TestInstanceProduct* 算法步骤描述如下:

1) 生成基准执行方案: 取  $A1, A2, \dots, Ak$  的一个隶属于有效类的数据值, 生成基准执行方案。

2) 有效类执行方案: 包括基准执行方案, 在基准执行方案的基础上, 逐个替换  $A1, A2, \dots, Ak$  的有效类中的数据值<sup>[10]</sup>, 生成全部的有效类执行方案。

3) 无效类执行方案: 包括基准执行方案, 在基准执行方案的基础上, 逐个替换  $A1, A2, \dots, Ak$  的无效类中的数据值, 生成全部的无效类执行方案。

4) 边界类执行方案: 包括基准执行方案, 在基准执行方案的基础上, 逐个替换  $A1, A2, \dots, Ak$  的边界类中的数据值<sup>[11]</sup>, 生成全部的边界类执行方案。

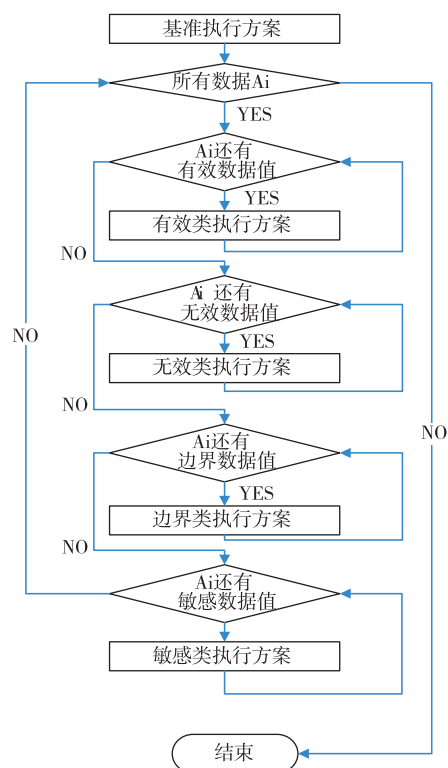


图3 *TestInstanceProduct* 算法流程图

5) 敏感类执行方案: 包括基准执行方案, 在基准执行方案的基础上, 逐个替换  $A1, A2, \dots, Ak$  的敏感类中的数据值, 生成全部的敏感类执行方案。

6) 根据上述算法描述, 测试用例设计系统自动生成的执行方案的个数公式如下:

$$SUM(implement) = 1 + \sum_{i=1}^k (N_i - 1) \quad (4)$$

### 3 测试用例设计模型的开发与应用

1.1 节和 1.2 节介绍了测试用例设计模型的概念、结构和表示方法,测试用例设计模型需要具有丰富测试经验的专业测试人员和测试技术专家按照一定的流程<sup>[12]</sup>进行开发。测试用例设计模型的开发过程如图 4 所示。首先从测试资产库中收集软件文档、测试项和测试用例等测试资产作为素材,然后分析测试用例模型设计素材,进行测试资产预处理,包括需求特征、测试类型和测试用例要素的分析、提取、归纳和总结<sup>[13]</sup>,继而按照测试用例设计模型的结构和表示方法,进行模型索引、模型基本信息、模型数据体、模型测试规程体和模型测试执行体的开发,测试用例设计模型开发完毕后,按照测试用例设计模型要求和有关标准、规范,进行测试用例设计模型的审核,审核通过后将测试用例设计模型存入测试用例设计模型库。要保证测试用例设计模型开发质量,有两点非常重要,一是测试用例设计模型的开发人员必须是测试技术专家,必要时领域专家也需要参加,二是开发工作必须按照测试用例设计模型的结构和内容实施,并进行严格审核。

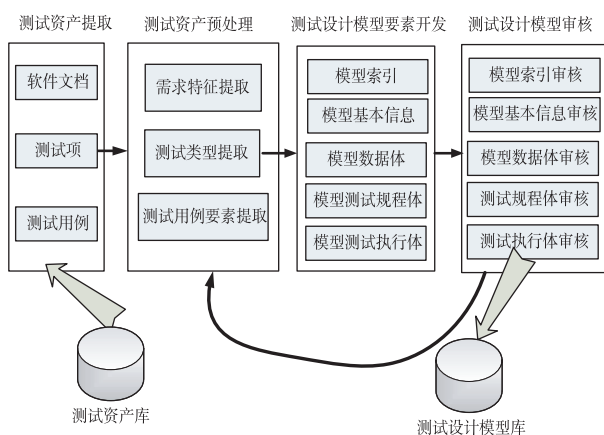


图4 测试用例设计模型开发过程

测试用例设计模型的引入和开发,主要目的是在测试用例设计过程中通过应用模型提高测试用例设计的规范性、有效性和效率。在测试设计过程中基于模型和基于人工方式的测试用例设计模式比较如图 5 所示。在基于人工方式的测试用例设计模式下,测试设计人员直接参与测试用例设计工作,是测

试设计工作的主体。在基于测试设计模型的测试用例设计模式下,测试用例设计模型是测试设计工作的主体,测试设计人员主要负责测试设计模型的开发和维护。

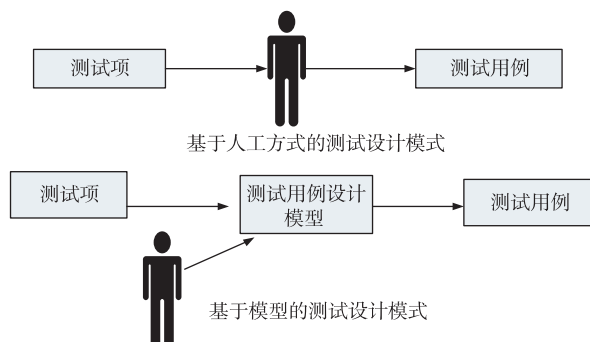


图5 基于人工和模型的测试用例设计模式

测试用例设计模型的应用流程如图 6 所示。首先从测试项库中提取测试项做为测试设计输入,然后提取需求特征、测试类型、数据项和其他要素,以测试项分析结果作为输入,基于测试用例设计模型自动完成测试用例生成,包括模型匹配、数据输入、模型各要素运行、测试用例生成、测试用例审核,审核通过的测试用例入测试用例库,不合格的重新进行测试设计。

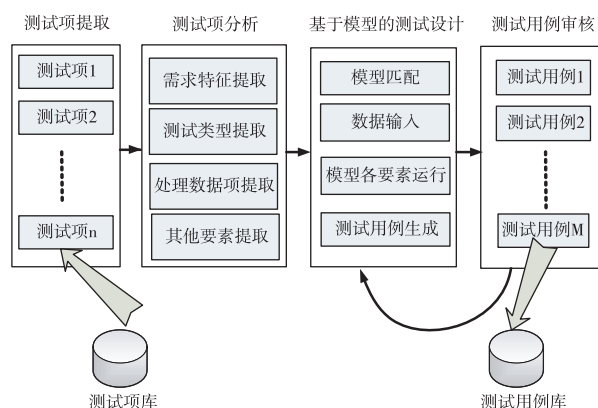


图6 测试用例设计模型应用流程

### 4 测试用例设计模型评价

以数据录入功能为例,软件测试技术专家根据积累的测试用例库中的该功能测试用例,开发出数据录入功能测试用例设计模型并通过审核。下面从测试设计有效性和测试设计效率两方面评价该模型。

首先评价该模型的测试设计有效性。设测试设计有效性量化值达到 1 表示针对数据录入功能测试用例设计结果是有效的,满足测试设计质量要求。软件测评机构分别组织测试人员测试了 S1、S2、S3、S4、S5 等 5 个软件,这 5 个软件都有部分数据录入功能。如果项目组未利用数据录入测试用例设计模型进行测试设计,而是由不同的测试人员分别进行 S1、S2、S3、S4、S5 软件数据录入功能的测试用例设计,通过评审不同测试设计人员设计的数据录入功能测试用例得出不同测试设计人员的测试用例设计结果质量的量化结果存在差异。如果每个项目组均以数据录入测试用例设计模型为依据进行测试用例设计,假定测试用例设计模型通过专家审核,则测试用例设计质量的量化结果为 1,是恒定的。所以,测试用例设计模型的测试设计有效性高于人工测试设计的有效性,评价示意图如图 7 所示。

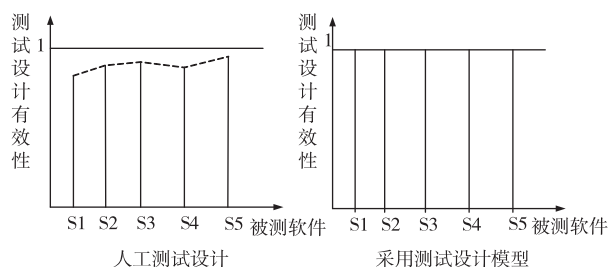


图 7 基于模型与人工的测试设计有效性评价

其次评价测试用例设计模型的效率。设采用测试用例设计模型进行该功能的测试用例设计所花费的时间成本为  $DesignTime_{model}$ , 人工进行该功能测试设计所花费的时间成本为  $DesignTime_{manual}$ , 则  $DesignTime_{model} = DevelopTime + DesgnRealTime$ , 其中  $DevelopTime$  为模型开发时间,  $DesgnRealTime$  为测试设计实际时间。由于采用测试设计模型自动进行测试设计,则  $DesgnRealTime$  是毫秒级的,非常小,可以忽略不计,  $DesignTime_{model} \approx DevelopTime$ 。如果人工测试设计,则  $DesignTime_{manual} = DesignTimeAve \times TestCaseNum$ , 其中  $DesignTimeAve$  为人工设计该用例的平均时间,而  $TestCaseNum$  为测试用例数量。从上述公式可以得出,  $DesignTime_{manual}$  是  $TestCaseNum$  的线性增函数。由此得出,  $TestCaseNum$  在一定范围内,可能  $DesignTime_{manual}$  会小于  $DesignTime_{model}$ , 但随着  $TestCaseNum$  的增长,  $DesignTime_{manual}$  肯定会超过  $DesignTime_{model}$ 。测试设计的效率评价如图 8 所示。

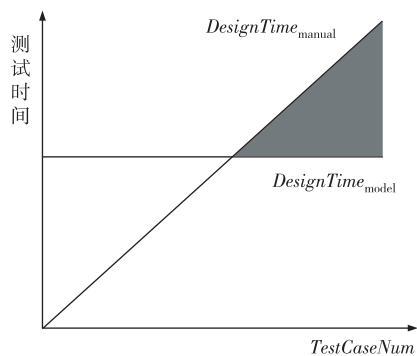


图 8 基于模型与人工的测试设计效率评价

## 5 结论

人工开展测试用例设计很难保证测试设计规范性、有效性和效率,因此本文深入分析软件测试过程和软件测试用例设计流程,以同类需求特征的测试用例设计方法相似为实践基础,提出了基于需求特征的测试用例设计模型,给出了模型的组成结构、表示方法、模型实例、模型开发与应用和模型评价。

测试用例设计模型具有动态性、知识性、规范性和易维护四个特点,模型的动态性体现在它会根据测试项输入的变化调整模型各组成要素的输出,该特性使得测试用例设计模型的应用范围大大增加。测试用例设计模型的知识性体现在它总结了测试设计专家的测试设计经验、技巧和方法。测试用例设计模型为测试资产的开发利用提供了重要基础和技术手段。测试用例设计模型的规范性体现在它主要部件的输出严格按照测试用例格式规范要求生成。其易维护体现在模型主要部件的开发和维护同高级程序设计语言的开发和维护相同,具有编程经验的测试人员能够开发和维护高质量的测试用例设计模型。测试用例设计模型的这 4 个特点为其应用和测试用例自动化生成奠定了重要基础。

## 参考文献

- [1] Tsai W, T Volovik D, Tkeefe T F. Automated test case generation for programs specified by relational algebra queries. *IEEE Trans on software Engineering*, 1990, 16(3): 316-324.
- [2] Weyuker E, Goradia T, Singh A. Automatically generating test case data from a boolean specification. *IEEE Trans on Software Engineering*, 1994, 20(5): 353-363.
- [3] 兰毓华, 毛法尧, 曹化工. 基于 Z 规格说明的软件测试用例自动生成. *计算机学报*, 1999, 22(9): 963-969.
- [4] 黄隲, 陈致明, 于洪敏等. 基于 UML 的软件测试用例

- 自动生成技术研究. 计算机应用与软件, 2004, 21 (11):16-18
- [ 5 ] 杨芙清,梅宏编著:面向复用的软件资产与过程管理. 北京:清华大学出版社,2008. 173-185
- [ 6 ] Marnie L. H. Software Testing Fundamentals: Methods and Metrics. New Jersey:Wiley,2003. 181-196
- [ 7 ] Glenford J M. The Art of Software Testing( Second Edition). New Jersey: John Wiley & Sons, Inc. 2004. 251-273
- [ 8 ] Tracey N, Clark J, Mander K, et al. Automated test-data generation for exception conditions. *Software Practice and Experience*, 2000, 30 ( 1 ) :61-79
- [ 9 ] Boris B. Software Testing Techniques ( Second Edition ). New York: International Thomson Computer Press, 1990. 86-101
- [ 10 ] Gelperin D. , Hetzel B. The Growth of Software Testing. *Communications of ACM*, 1988, 31(6) :687-695
- [ 11 ] 柳纯录,黄子河,陈涿萍. 软件测评师教程. 北京:清华大学出版社,2005. 208-221
- [ 12 ] Illes T, Herrmann A, Paech B, et al. Criteria for Software Testing Tool Evaluation-A Task Oriented View. In: Proceedings of the 3rd world Congress for Software Quality. Munich, Germany, 2005. 506-516
- [ 13 ] Horwath T. Automated Test Tools Evaluation Criteria. Version 1.02, January 2007

## Study of a test case design model for software based on characteristic of test requirement

Li Wenwu \*, Li Wenfa \*\*, Duan Miyi \*

( \* School of Computer Science and Engineering, Beihang University, Beijing 100191 )

( \*\* College of Information Technology, Beijing Union University, Beijing 100101 )

### Abstract

In view of the stability and efficiency problem in software test design under the instance that different testers design a test case for the same characteristic of test requirement, a test case design model based on the characteristic of test requirement is proposed. The test case design model is a set that includes five members as  $\langle ModelIndex, ModelInfo, TestDataBody, TestProBody, TestRunbody \rangle$ . *ModelIndex* is the model index, *ModelInfo* is the model basic information, *TestDataBody* is the model data set, *TestProBody* is the model regulation body, and *TestRunbody* is the model run body. The test design model has four characteristics of self-adaption, normative, knowledgeable, and convenient to develop. The results of the evaluation performed according to the stability and efficiency of software test design show that the model is very important to improving the stability and efficiency in software test design.

**key words:** requirement characteristic, test case design model, developing, evaluation