

# 测试用例自注册与按需执行系统<sup>①</sup>

张金<sup>1,2</sup>, 李茹<sup>1</sup>, 胡先兵<sup>2</sup>

<sup>1</sup>(华中科技大学机械科学与工程学院, 武汉 430074)

<sup>2</sup>(武汉开目信息技术有限责任公司, 武汉 430223)

**摘要:** 测试用例是测试工作中的核心内容。在实际测试项目中, 测试用例的管理维护是测试环节中的重要部分。用例管理工具的功能是影响测试工作质量与效率的重要因素。通过对当前测试用例管理工具的分析, 提出一个可实现用例信息自注册、用例按需自动执行的管理系统

**关键词:** 测试用例; 测试用例管理; 用例信息注册; 用例按需执行

## Test Case Self-Register and Execute On-Demand System

ZHANG Jin<sup>1,2</sup>, LI Ru<sup>1</sup>, HU Xian-Bing<sup>2</sup>

<sup>1</sup>(College of Mechanical Science & Engineering, Huazhong University of Science & Technology, Wuhan 430074, China)

<sup>2</sup>(Kaimu Information Technology Ltd, Wuhan 430223, China)

**Abstract:** Test case is the central content of the test progress. In the actual test items, test case management and maintenance are important parts of the test session. The function of the case management tool is an important factor in the impact of testing quality and efficiency. Based on the analysis to current test case management tools, we give a system which can realize test case information self-register and case execute on-demand.

**Keywords:** test case; test case management; case information register; case execute on-demand

## 1 引言

对于软件开发来说, 测试用例的重要性毋庸置疑, 它是软件测试的核心, 是软件产品质量保证的基础。随着软件功能的增多和增强, 软件测试人员需要编写并执行大量测试用例来检验产品功能和性能的可靠性。这就导致被测产品的测试用例数量非常庞大<sup>[1]</sup>, 日常测试中就会面临以下两种情况: (1) 用例若不能被很好地收集, 很容易漏测和丢失一些用例, 导致测试质量下降和重复编制测试用例的问题; (2) 用户在验证缺陷修复与回归测试中, 常常需要从测试用例全集中根据需要挑选某些测试用例来做测试, 这对于测试人员来说是一项繁琐复杂的工作。因此我们需要研究如何在保证充分测试的前提下, 生成尽可能小的用例集来进行软件测试, 有效提高测试效率<sup>[2]</sup>。

为了解决测试过程中用例管理的问题, 用例管理工具应运而生。现有的主流管理工具可以解决用例管

理中的很多问题, 但也或多或少存在一些不足: 例如 TestManager 具有功能强大, 可以对测试用例无限分级, 而且可以和 Rational 的测试工具 robot、functional 相结合等优点, 但在管理测试用例时不太稳定, 有时会造成测试用例的丢失。Wiki 具有交流方便, 有版本控制、比较功能等优势, 但测试用例的统计不方便, 需要编写专门的程序。

在对现有工具分析的基础上, 研究一个测试用例的管理系统, 目的是实现用例向用例库中自动注册以及根据需求配置自动依次执行, 以解决现有工具中存在的用例丢失、难找、统计不方便等问题。进一步规范软件测试用例管理的方法和途径, 使当前的测试用例管理过程更加合理化和自动化。通过按功能、按模块分类, 使得模块编程负责人员、测试人员双方共同维护和管理测试, 同时发挥接口兼容性和稳定性的测试效果, 兼顾开发人员接口的全面性设计和维护与测试。

<sup>①</sup> 收稿时间:2010-04-21;收到修改稿时间:2010-05-16

## 2 系统的体系结构

如图 1 所示, 测试用例自注册与按需执行系统主要由用例注册模块、测试计划管理模块、用例按需自动执行模块组成。其中, 用例注册模块将用例基本信息自动收集注册到数据表中, 以便测试人员可以直观了解测试用例情况, 基本信息包括用例名、所属模块、创建时间等; 测试计划管理模块是用来制定测试计划、新增测试计划、删除测试计划、修改测试计划等, 为更好更方便的选择用例、执行用例服务; 用例按需自动执行模块, 系统识别被选中的用例, 自动跳转至用例代码段执行测试功能。

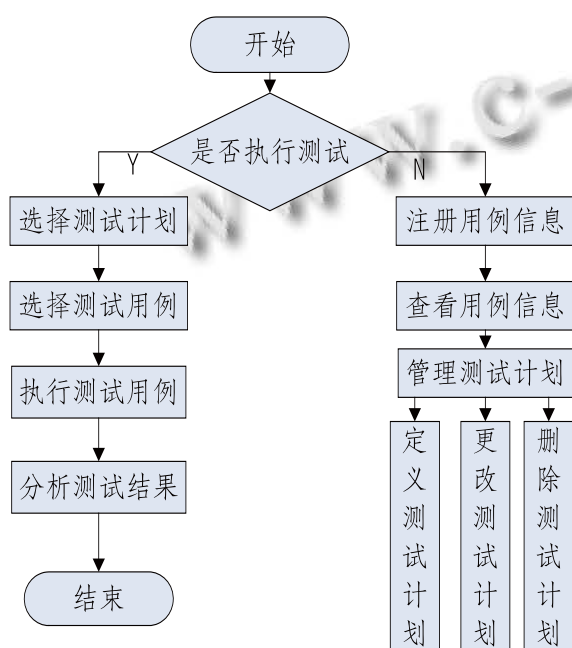


图 1 系统执行流程图

## 3 系统实现关键技术及策略

不同类型的用例有各自的特点。为实现系统的功能, 有几个关键技术需要解决: (1) 用例数量非常大, 怎样便捷地把所有用例信息无遗漏的抓取出来; (2) 为了能够更快的选择所需用例, 需要对测试计划进行管理维护; (3) 选择好用例以后, 怎么样实现自动执行。下面就这三个关键问题进行分析并提出解决方案。

### 3.1 用例信息自注册及用例执行

#### 1) 用例信息收集并注册

不断地开发与测试积累了大量用例, 每个用例都有自己的特征及针对性。测试人员面对如此大量的用

例, 如何在茫茫代码中快速获得所需用例信息成为问题的关键。如果能够提供一种快速直观的获得每个用例的信息的方法, 对于了解、选择、执行用例都将带来便捷。因此, 为该系统设计了用例信息自注册功能, 该功能可将用例库中的用例信息全部抓取出来, 按规则集中存放, 显示于界面上, 这样, 测试人员便可以一目了然看到测试用例的基本情况, 包括有多少模块, 每个模块下有多少用例, 以及各模块各用例的属性等信息。

为了能将每一个用例的信息都提取出来而不产生遗漏的情况, 系统规定了一种统一的用户录入格式, 管理人员在添加用例时, 按照既定的格式进行输入。用例添加的格式为用例头部注明用例基本信息, 测试体为用例实体部分。以数据库操作程序的测试用例为例:

```
if(Askmode("数据库操作"))//模块名称
{
    if(AskCase("更新数据","数据库","09-1-1","2.1"))
        //用例基本信息提取及自动注册
    {
        // 更新数据
        CString score2=
        GetFieldValue("Y","Grade",1);
        ... //剩余部分代码段
    }
    if(AskCase("删除数据","数据库","09-3-2","2.2"))
    {
        //删除指定行数据
        DeleteRow("Y",2,NULL);
        SaveData("Y",0,0);
    }
    ... //后续测试用例
}
```

程序启动后, 通过执行 Askmode()和 AskCase()函数并返回 false 不执行测试体, 而只将用例信息提取出来, 注册到信息表中, 供集中管理维护, 并提供模糊搜索等方便、快捷功能。如果发现有新增加、修改或删除这些特殊状态用例存在, 会自动同步注册信息。

#### 2) 用例实体执行

一旦用例信息被提取, 就可以根据信息选择用例,

然后通过控制这些用例执行时 Askmode()和 AskCase()函数返回 true 来执行用例主体, 实现选择所需测试用例自动执行的功能。下图为用例注册与执行过程流程示意图:

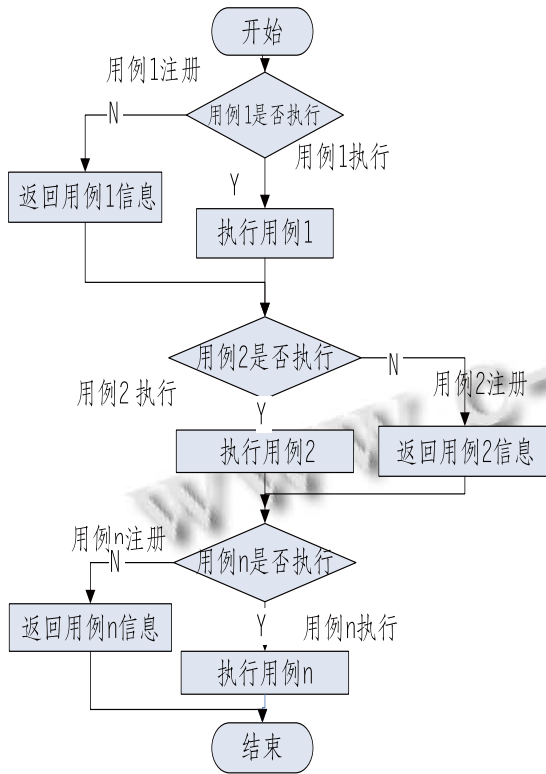


图 2 程序运行流程图

### 3.2 测试计划管理

为了能从数量庞大的测试用例中精心挑选所需的测试用例, 使之与测试数据配合覆盖全部路径、达到最佳的测试效果, 必须精心设计测试用例, 并提供高效的用例管理方式。而软件开发中有大量的测试模块和用例, 如果将所有的这些信息都全部显示出来, 就为用户挑选用例造成了麻烦。所以有必要制定一些测试计划, 按照某种原则定义一些配置方案, 例如哪几个模块是经常一起被运行的, 哪些用例可以测试某一功能等, 经用户定义, 放在一个测试计划中。测试计划的名称要达到见文知意的效果, 即用户看到测试计划, 就可以大概了解它是测试什么功能。比如“表缓存函数功能测试模块”, 在执行测试工作时, 用户首先选择某一个测试计划, 主界面中将只显示该计划下的模块和用例, 而不是所有模块和用例。另外, 不同的测试计划项中可以有

相同的模块以及用例。

通过制定测试计划, 测试人员不必面对数目繁多的模块和用例, 根据测试对象的不同, 选择不同的配置方案, 使用例的显示、选择以及执行显得条理清晰, 目的性强, 效率高。随着更多的开发, 可能会淘汰一些用例, 也可能根据实际情况修改以前的用例, 还会不断产生新的用例, 所以测试计划也要随之变动。因此, 测试计划可以被更新、删除、添加。

对于维护测试计划, 系统也提供了较为方便的方式。在功能定义部分, 界面中将显示所有的模块及其所属的用例信息, 使用者根据需要, 选择所要制定计划中包含的模块及用例, 确定以后, 就会出现配置界面, 供使用者新建、删除、修改测试计划。下图为测试计划管理界面。

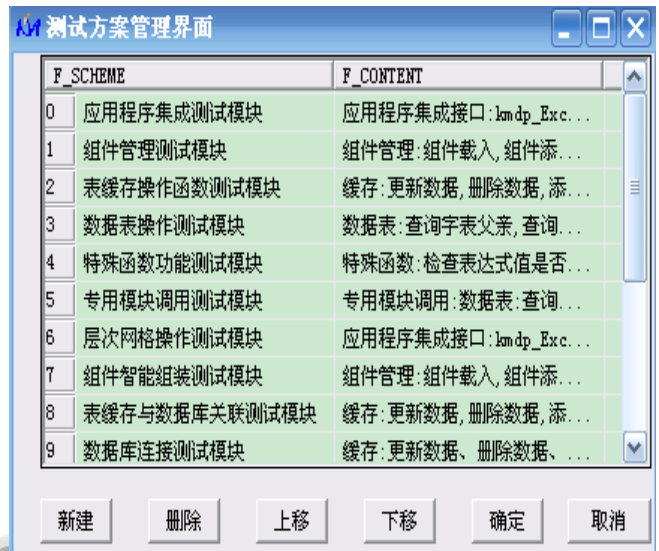


图 3 测试计划管理界面

### 3.3 根据需求自动执行用例功能的实现

#### 1) 用例自动执行

用例的自动执行是该系统的重要功能之一, 前面论述的无论是用例注册还是测试计划的管理都是为该功能服务的, 也就是说最终目的是更好地执行测试用例。通过以下三个步骤实现用例灵活选择及按需自动执行: (1) 识别被选中的用例, 选择一个测试计划进入, 界面中将显示该方案下的模块及用例信息, 模块与用例的属性中都有“是否选择”项, 用户可以在上面任意勾选一个或多个模块、用例, 系统会判断被选中的项目; (2) 选中的用例属性信息依次与程序中的代码段

匹配,找到关键字相同的项,定位于被选中用例;(3)定位以后,进入用例代码段,执行用例。下图 4 是一个筛选用例的界面,显示了选择某一测试计划后进入的界面。



图 4 选择以及自动执行测试用例界面

由图所示,选择特定的测试计划大大缩小了用例的范围,但是用例数量仍然很多,如果还需要进一步缩小范围,可以使用基于版本和日期的筛选。用户选中版本号 and 日期以后,版本及日期大于选中值的用例被过滤出来显示到界面上。为用户选择用例提供了方便,查找更加快速。

## 2) 测试报告的生成

测试过程中,每个用例测试一个重点功能,且每个

用例的测试重点都有相应注释。选中的用例执行结束后,系统会自动生成一个测试报告文档,该文档记录用例的运行结果和测试结论。

## 3.4 小结

测试用例的管理维护是一个不间断的过程,该系统为更好的维护和管理测试用例提供了可行性,实现了用例信息的自动注册以及根据需要自动选择执行用例,有效地解决了用例丢失的问题,同时使得用例执行更加灵活,极大地提高了测试效率。此外,在实际测试过程中,软件功能以及实现可能发生改变,导致用例信息不断的更新。系统实现了用例信息的动态更新及维护。

## 4 结束语

软件测试是保证软件质量的重要手段,而测试用例管理工具的性能决定测试工作的效率。本文以实际项目为背景,对当前软件测试管理工具进行了研究分析,结合实际需求,给出了一个软件测试管理系统的解决方案,设计开发了一个自注册式测试用例管理与自动执行系统,使得测试用例的管理更加高效、方便,减少了用例管理人员的工作量,同时也提高了用例执行的灵活性和效率。目前该系统已被应用于实际的测试工作中,且运行状态良好。

## 参考文献

- 1 朱海燕,范辉,谢青松,刘培强.测试用例排序的研究.计算机工程与科学,2008,30(1):79—80.
- 2 陈伟,唐红光.基于测试需求集的测试用例集极小化算法.人工智能识别技术,2007:811—812.