面向函数调用路径的测试技术与方法

牟永敏 著

清華大學出版社

面向函数调用路径的测试技术与方法

牟永敏 著

清華大學出版社
北京

内 容 简 介

本书介绍了“面向函数调用路径测试技术和方法”的相关概念、技术和方法，结合当前软件开发的两种模式：面向过程和面向对象的技术特点，提出了面向过程和面向对象两种函数调用路径静态提取的技术方法，建立以此为基础的路径测试策略和测试模型，有效解决了集成测试、系统测试，特别是回归测试中，测试用例的有效性问题，包括：不可达路径的检测、动态测试路径拆分、测试过程可视化跟踪、软件变更可视化跟踪、回归测试用例精简等。

本书可以供从事软件测试研究的高校教师、企业软件测试高层管理者、软件测试科研人员参考，也可作为高等院校计算机相关专业研究生的教材和参考书。

图书在版编目(CIP)数据

面向函数调用路径的测试技术与方法/牟永敏著. --北京：清华大学出版社，2014
ISBN 978-7-302-36544-0

Ⅰ. ①面…　Ⅱ. ①牟…　Ⅲ. ①软件－测试－研究　Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2014)第 105577 号

责任编辑：付弘宇　薛　阳
封面设计：迷底书装
责任校对：焦丽丽
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：17.25　　**字　　数**：419 千字
版　　次：2014 年 8 月第 1 版　　**印　　次**：2014 年 8 月第1 次印刷
印　　数：1～1000
定　　价：59.00 元

产品编号：057601-01

前　言

信息化时代的主要表现就是计算机软件的广泛应用，随着软件应用领域和规模的不断扩大，软件的复杂程度也不断提高，软件质量和可靠性等问题更加突出。用户希望买到安全可靠的软件，以提高工作效率、保证工作质量；软件开发商希望开发出高质量软件产品，以减少开发和维护成本。

“很多人都认为微软是一家软件开发公司，而事实上我们是一家软件测试公司。在最后产品要发布的时候，此产品的所有相关部门都必须签字，而测试人员具有绝对的否决权”——微软公司总裁——比尔·盖茨。这段话充分表明：软件测试是保证软件质量的重要手段。为此，很多国内外大型软件开发机构都建立自己的软件测试部门，软件测试从业人员大量增加，软件测试已成为一个蓬勃发展的行业。

随着软件测试的发展，软件测试技术和方法也在不断更新。如何提高软件测试充分性，使得软件测试工作更加准确完整，是软件测试专家学者广泛讨论的热点问题。作者总结多年的教学和科研成果，提出了“面向函数调用路径的测试技术和方法”。我们知道一个系统无论多么复杂，它都要完成用户的一个个任务，而这些任务是由一个个功能组合而成的，这就构成了一条函数调用路径 FCP(Function Calling Path)。我们把一个系统看成是由有限条函数调用路径组成的，如果对所有的函数调用路径都进行了充分测试并且无误的话，那么这个系统是可靠的。“面向函数调用路径的软件测试方法”，简化了面向基本路径的软件测试问题，使得面向基本路径的测试工作量呈指数级降低，将面向路径的测试由不可能变成了可能。

软件的迭代开发需要更改部分代码，包括：数据更改、控制逻辑更改、功能实现算法修改、功能点的增删等。为了验证修改后的代码是否产生了新的问题，在每次修改后，都必须进行回归测试，而回归测试测试用例的选择是软件测试的难点，全面测试必然造成冗余测试工作量，不全部测试难以保证回归的充分性。书中提出了“面向函数调用路径变更影响域的分析策略和方法”，测试人员可以先确定变更影响域，进而确定变更影响域下的测试用例，有效约简了测试用例数量，显著提高了回归测试效率。

相关科研项目“软件测试新技术和方法研究”，荣获 2009 年度中国人民解放军科技进步二等奖；“软件变更影响及安全性测试技术与应用”，荣获 2013 年北京市科技进步三等奖。所研发的软件测试工具具有广阔的应用前景。

全书共分 10 章。第 1 章讲述软件测试的基础知识，着重介绍了传统的软件测试的基本思想和原理，以及近年来逐渐流行起来的软件测试技术和方法的主要优点；第 2 章阐述面向过程的函数静态调用路径生成的理论和方法，包括面向函数调用路径的基本概念，函数调用路径的提取方法等；第 3 章阐述面向对象的函数静态调用路径生成的理论和方法，包括：

面向对象语言重载、模板、多态等特性情况下的唯一路径确定技术和方法；第4章阐述面向函数调用关系的覆盖分析方法，包括动态路径拆分、组合、约简等技术和方法；第5章阐述面向函数调用关系的测试用例优化，包括测试用例的约简、优化等；第6章阐述面向函数调用关系的测试用例自动生成，包括自动生成原理和实现方案等；第7章阐述面向函数调用关系的变更影响分析，包括：回归变更影响域的确定策略和方法、回归测试用例确定、优先级排序等；第8章介绍了Visual Studio依赖图生成工具，让读者了解当前流行的测试工具；第9章介绍了项目组自己研发的Regression Test for C 2.0工具；第10章介绍了DTS工具。

本书在注重介绍“面向函数调用路径测试技术和方法”相关理论的同时，尽量提供一些简洁、实用和易实现的算法，让读者在学习了本书以后，能解决在软件测试领域中遇到的相关问题。

第1～9章由牟永敏、王锐、董伟娜、齐胜、丁媛、朱绪利、李良杰、刘梦婷、张亚楠、李慧丽、刘昂、杨志嘉、白倩倩、范浩杰、姜志超编著。第10章由宫云战、张大林编著。刘丹凤、徐爱平、郑锦勤、闫明明、徐明刚对书稿进行了校对。最后由牟永敏统一修改定稿。由于时间仓促，作者水平有限，书中欠妥和纰漏之处在所难免，恳请读者和同行批评指正。

编　者

2014年4月

目　　录

第1章 绪论

随着社会的不断进步，软件在社会生活中起着越来越重要的作用。软件开发的各个阶段都需要人的参与，因此，出现错误是在所难免的。并且随着计算机所能控制的对象复杂度的不断增强和软件功能的不断增强，软件的规模也在不断增大，这使得错误更可能发生。软件错误的存在有可能造成巨大的经济损失，甚至危害人的生命安全。软件质量已经成为制约计算机应用的主要因素之一。软件测试作为软件开发过程中的一项重要工作，是保证软件质量的重要手段[1]。

1.1 软件测试

软件生命周期是软件的产生直到报废的生命周期。周期内有问题定义、可行性分析、总体描述、系统设计、编码、调试和测试、验收与运行、维护升级到废弃等阶段，这种按时间分层的思想方法是软件工程中的一种思想原则，即按部就班、逐步推进，每个阶段都要有定义、工作、审查、形成文档以供交流或备查，以提高软件的质量。测试工作在软件开发的整个过程中占有极其重要的地位，其工作量通常占软件开发总工作量的40%～60%。在开发Windows 2000的过程中，除了1700多个开发人员外，其内部的测试人员就有3200人。开发和测试人员之比大约为3∶5。很多专家认为，这才是一个成熟的软件产品在制造过程中人员的合理比例。因此软件测试在软件生命周期中是一项非常重要的工作。

1.1.1 软件测试的目的与意义

在进行软件测试之前，首先需要明确软件测试的目的是什么。测试的目的是要以最少的人力、物力和时间找出软件中的各种错误与缺陷，通过修正各种错误与缺陷来提高软件质量，避免软件发布后由于潜在的软件缺陷和错误造成的隐患所带来的经济风险。同时，测试是以评价一个程序或系统属性为目标的活动。测试是对软件质量的量度与评价，以验证软件的质量满足用户需求的程度，为用户选择与接受软件提供有力的依据。

此外，通过分析错误产生的原因还可以帮助发现当前开发工作所采取的软件过程的缺陷，以便进行软件过程的改进。同时通过对软件结果的分析整理，为风险评估提供信息，还可以修正软件开发规则，并为软件可靠性分析提供依据。当然，通过最终的验收测试，也可以证明软件满足了用户的需求，树立人们使用软件的信心。

GBT 15532—2008计算机软件测试规范给出了软件测试的三个目的：

(1) 验证软件是否满足开发合同或项目开发计划、系统/子系统设计文档、软件需求规格说明、软件设计说明和软件产品说明等规定的软件质量要求;

(2) 通过测试,发现软件缺陷;

(3) 为软件产品的质量测量和评价提供依据。

软件测试的这三个目的是指导我们进行软件测试过程的重要依据。开发软件的目的是使用户完成预定的任务,并满足用户的需求。因此,软件测试要验证软件是否能满足以用户需求为基准的各种文档的要求。同时,测试的过程就是不断地发现缺陷,促使开发人员修复缺陷,从而使软件质量能够得到有效提升的过程。因此软件测试要尽可能地找出软件中存在的缺陷,尽早地修复这些缺陷,以避免软件缺陷所带来的巨大损失。同时通过对软件测试过程中发现的缺陷进行分析和统计,可以帮助项目成员改进其工作,也可以通过这些分析来预测软件中潜在的缺陷数及严重程度,从而为软件产品的质量测量与评价提供依据。

1.1.2 软件测试分类

按照不同的分类方法,软件测试可分为不同的种类,如图 1.1 所示。

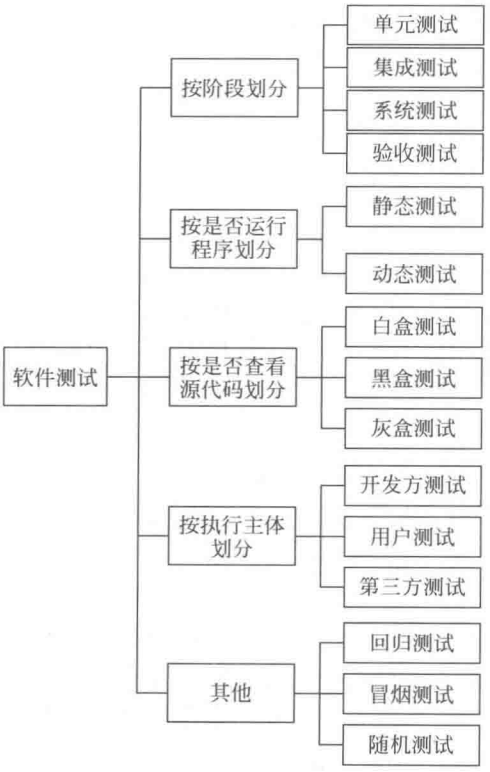

图 1.1 软件测试分类

1. 按照执行状态分类

1) 静态测试(Static Testing)

静态测试指不运行程序,而通过人工或利用自动检测工具对程序和文档进行分析与检查。静态测试技术又称为静态分析技术,是对软件中的需求说明书、设计说明书、程序源代

码等进行的非运行检查。静态测试包括代码走查、代码审查等。

2）动态测试(Dynamic Testing)

顾名思义，是指实际运行被测程序，输入相应的测试数据，检查实际输出结果和预期结果是否相符的过程，所以判断一个测试是属于动态测试还是静态测试，唯一的标准就是看是否运行程序。

2. 按测试方法划分

1）白盒测试(White-Box Testing)

白盒测试也称结构测试或逻辑驱动测试，是指基于一个应用代码的内部逻辑知识，即基于覆盖全部代码、分支、路径、条件的测试。它指导产品内部的工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都能按预定的要求正确工作。

2）黑盒测试(Black-Box Testing)

黑盒测试是指不基于内部设计和代码的任何知识，而基于需求和功能的测试。黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能的前提下，通过测试来检测每个功能是否都能正常使用的。在测试时，把程序看作一个不能打开的黑盒子。在不考虑程序内部结构和内部特性的情况下，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，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）灰盒测试(Gray-Box Testing)

灰盒测试技术是一种有效的、介于白盒测试与黑盒测试之间的技术，它既关注程序运行时的外部表现，又注意程序内部高层逻辑结构。灰盒测试的优点是测试结果可以对应程序的内部粗/细粒度路径，便于缺陷的定位、分析和解决。

软件测试方法的技术分类与软件开发过程相关联，单元测试一般使用白盒测试方法，集成测试使用灰盒测试方法，而系统测试和验收测试则使用黑盒测试方法。

3. 按开发阶段划分

1）单元测试(Unit Testing)

单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等。它是软件动态测试最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。因为单元测试需要知道内部程序设计和编码的细节知识，一般应由程序员而非测试员来完成，往往需要开发测试驱动模块和桩模块来辅助完成单元测试。

2）集成测试(Integration Testing)

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单元之间的接口是否正确。它根据集成测试计划，一边将模块或其他软件单元组合成越来越大的系统，一边运行该系统，以分析所组成的系统是否正确，各组成部分是否合拍。集成测试的策略主要有自顶向下和自底向上两种。

3）系统测试(System Testing)

系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的功能和性能

等满足其需求说明所指定的要求。检查软件的行为和输出是否正确并非一项简单的任务，它被称为测试的“先知者问题”。因此，系统测试应该按照测试计划进行，其输入、输出和其他动态运行行为应该与软件需求说明进行对比。

4）验收测试(Acceptance Testing)

验收测试旨在向软件购买者展示该软件系统满足其用户的需求。它的测试数据通常是系统测试的测试数据的子集。所不同的是，验收测试常常有软件系统的购买者代表在现场，甚至是在软件安装使用的现场。这是软件在投入使用之前的最后测试。

4. 按照执行主体划分

1）开发方测试

开发方测试通常也叫“验证测试”或“α测试”，开发方通过检测和提供客观证据，证实软件的实现是否满足规定的要求。验证测试是指在软件开发环境下，由开发者检测与证实软件的实现是否满足软件设计说明或软件需求说明的要求。

2）用户测试

用户测试是指在用户的应用环境下，用户通过运行和使用软件，检测与核实软件实现是否符合自己预期的要求。通常情况下的用户测试不是指用户的“验收测试”，而是指用户的使用性测试，由用户找出软件的应用过程中发现的软件缺陷与问题，并对使用质量进行评价。β测试通常被看作一种“用户测试”。β测试主要把软件产品有计划地免费分发到目标市场，让用户大量使用，并评价、检查软件。通过用户各种方式的大量使用，来发现软件存在的问题与错误，把信息反馈给开发者修改。β测试中厂商获取的信息，有助于软件产品的成功发布。

3）第三方测试

第三方测试是指介于软件开发方和用户方之间的测试组织的测试。第三方测试也称独立测试。软件质量工程强调开展独立验证和确认活动。软件第三方测试就是由在技术、管理和财务上与开发方相对独立的组织进行的软件测试。

5. 其他测试

1）回归测试(Regression Testing)

回归测试是指对软件的新版本测试时，重复执行上一个版本测试时的用例。比如，我们测完了某软件，将其交给开发人员修改，开发人员修改后将新版本重新交给测试人员进行测试。这时测试员要将上一版本中所执行的测试有针对性地重复执行一遍，而不仅仅是测试上一个版本中出错的地方。因为软件的新版本中有可能引入新的缺陷，需要我们系统全面地重新测试。回归测试可以在任何测试阶段进行(单元测试、集成测试、系统测试、验收测试等)，既有黑盒测试的回归，也有白盒测试的回归。

2）冒烟测试(Smoke Testing)

冒烟测试是指在对一个新版本进行系统大规模的测试之前，先验证一下软件的基本功能是否实现，是否具备可测性。冒烟测试名字的由来与电路板的测试有一些关系。早期人们在测试一种型号的电路板质量时，先给它接上电，如果板子冒烟烧坏的话，就说明板子的质量存在严重问题，必须拿回去重新研制或生产。引入软件测试中，就是指测试小组在正规

测试一个新版本之前，先指派一两个测试人员测试一下软件的主要功能，如果这都没有实现的话，就返回开发组重新开发。这样做的好处是可以节省大量的成本。

3）随机测试（Random Testing）

随机测试是指测试中所有的输入数据都是随机生成的，其目的是模拟用户的真实操作，并发现一些边缘性的错误。有人也把随机测试叫做猴子测试：如果让一百万只猴子在一百万只键盘上敲一百万年，他们最终就有可能写出莎士比亚的巨著。当软件发布后，就会有成千上万的人像猴子一样对软件乱敲乱点。在软件发布之前，尽量模拟这种随机的操作，就有可能发现一些隐蔽的错误。

1.1.3　国内外软件测试现状

软件测试在软件较发达的国家（比如美国），已经发展成为一个独立的产业，主要体现在以下几个方面：

（1）软件测试在软件公司占有重要地位。在微软，一个项目组中测试工程师要远比编码工程师多。同样，花在测试的时间也比花在编码的时间多。

（2）软件测试理论研究蓬勃发展。每年举办的各种各样的测试技术年会，发表了大量的软件测试研究论文，引领软件测试理论研究的国际潮流。

（3）软件测试市场繁荣。美国有一些专业公司开发软件测试标准与测试工具，MI Compuware、MaCabe、Rational 等都是著名的软件测试工具提供商，它们出品的测试工具已经占领了国际市场。

我国的测试技术起步于“六五”，随着软件工程的研究而逐步发展起来。1990 年国家级的中国软件评测中心成立，测试服务逐步开展起来。

由于起步晚，因此无论在软件测试理论研究还是测试实践上，都和发达国家有较大的差距，主要体现在对软件产品化测试的技术研究还较为贫乏，从业人员少，测试服务没有形成足够的规模等方面。但是，随着我国软件产业的蓬勃发展及对软件质量的重视，软件测试也越来越被人们所看重，软件测试正逐步成为一个新兴的产业。

我国正迈入测试时代，主要体现在以下几个方面：

（1）我国著名软件公司都已经或着手建立独立的专职软件测试队伍。当然人员规模及比例无法和国外大公司相比，但毕竟在公司内部贯彻了独立测试的意识。

（2）国家人事部和信产部于 2003 年关于职业资格认证第一次在我国提出了“软件评测师”的称号，这是国家对软件测试职业的高度重视与认可。

（3）在信产部关于计算机系统集成资质以及信息工程系统工程监理资质的认证中，软件测试能力已经被定为评价公司技术能力的一项重要指标。

（4）2001 年，信产部发布的部长 5 号令，实行了软件产品登记认证制度，规定凡是在我国境内销售的产品必须到信产部备案登记，且要经过登记测试。

（5）自 2001 年起，国家质检总局和信产部都通过测试对软件产品进行质量监督抽查。

（6）国家各部委、各行业正在通过测试规范行业的健康发展，通过测试把不符合行业标准要求的软件拒之门外，对行业信息化的健康发展起了促进作用。

（7）用户对软件质量要求越来越高，信息系统验收不再走过场，而要通过第三方测试机构的严格测试来判定。

(8)“以测代评”正在成为我国科技项目择优支持的一项重要举措,如国家“863”计划对数据库管理系统、操作系统、办公软件、ERP 等项目的经费支持,都是通过第三方测试机构科学客观的测试结果来决定的。

(9) 软件测试正成为部分软件学院的一门独立课程,对我国软件测试人才的培养起到作用。

(10) 第三方测试机构得到了蓬勃发展。近两年全国各地新成立的软件测试机构有十多家,测试服务体系已经基本确立。

由上可见,我国的软件测试行业正处在一个快速成长的阶段。随着时间推移,我们与发达国家的差距必然会逐步缩小。

1.2 回归测试

在软件的生命周期中,软件的演化、软件需求的改变、技术的更新和软硬件的升级都会引起软件的变更。变更后的软件功能是否能达到预期?原有的软件功能是否被损害?这两个问题,将在回归测试过程得到回答。回归测试是一种验证已变更系统的完整性与正确性的测试技术,对保证软件质量具有重要意义。

1.2.1 回归测试概念

回归测试是指软件系统被修改或扩充后,为了保证对软件的修改达到了预期的目的并且没有引入新的错误而重复进行的测试。理论上,每当软件发生修改和变化时都要对原来已经通过测试的区域进行重新测试,即需要进行回归测试来验证修改的正确性及其影响。

GBT 11457—2006 信息技术、软件工程术语给出了一个回归测试的标准定义:回归测试是系统或部件选择的重新测试,用以验证修改未引起不希望的有害效果,或修改后的系统或系统部件仍满足规定的需求。

1.2.2 为什么要进行回归测试

在软件测试和软件维护过程中,为了证明程序的修改对程序的其他部分没有产生负面影响而进行的测试称为回归测试,也就是指修改了旧代码后,重新进行测试以确认修改代码后没有引入新的错误或导致其他代码产生错误。

在软件开发、维护和升级的不断演化过程中,由于各种各样的原因,例如,功能性/非功能性需求的变更、技术更新和软/硬件平台升级,使得软件系统经常发生改变。这些改变会给软件系统带来风险,因为改变传播效应可能会引入新的错误,有时甚至是致命的错误。

回归测试并不是只在需求变更时进行的,它可以发生在软件生命周期的任意一个部分,从单元测试,功能测试,集成测试,甚至到发布测试。事实上,渐进和快速迭代开发中频繁的回归测试可以更加有效地提高代码质量,从而缩短回归周期。

回归测试是针对软件产品的基线版本的改变所做的测试,因此通过对产品进行回归测试可以

(1) 验证开发人员所承诺修复的软件缺陷是否已被正确修复。

(2) 验证新的软件修改是否影响了原有的稳定模块的正确性和稳定性。

(3) 验证新的软件修改是否引入了新的缺陷。

(4) 验证新的软件版本是否稳定，从而成为新的基线版本，以备后续开发使用。

所以每当软件发生变化时，就必须重新测试现有的功能，以便确定修改是否达到了预期的目的，检查修改是否损害了原有的正常功能。同时，还需要补充新的测试用例来测试新的或被修改的功能。因此，为了验证修改的正确性及其影响就需要进行回归测试。

1.2.3　回归测试与其他测试的区别

回归测试作为软件测试的一种重要方法，贯穿整个软件周期，能使用软件测试的一般技术但又具有其自身的特点。软件测试过程首先是编写测试用例，配置测试环境，然后执行测试用例，对测试的结果进行评价，如果符合需求规格说明书就完成软件测试；如果有错误，就要对错误进行修改，修改后为了检验修改是否达到目标，需要进行回归测试。在软件的维护阶段，也会出现一些修改错误或添加新的功能，也同样需要回归测试。

回归测试的过程与软件测试中其他测试的过程不同，其测试计划的来源，测试的范围，分配的时间，所需的开发信息，完成的时间以及测试执行的频率等这几个方面都有很大的差异，表 1.1 是回归测试与其他测试在这几个方面的一个比较。

表 1.1　回归测试与其他测试的比较

	其 他 测 试	回 归 测 试
测试计划	已有的带有一些测试用例的测试计划，该测试用例从未被执行过	更改了的规格说明书、修改过的程序和一个需要更新的旧的测试计划
测试范围	过程目标是要检测整个程序(包括各个模块及模块之间的互相作用)的正确性	重点关心程序中被修改以及被修改影响的程序部分的正确性，对于没有修改且没有被修改影响的部分无须重新测试
时间分配	所需时间通常是在一个产品开发之前都被预算好的	所需的时间不包含在整个产品进度表和花费上
开发信息	关于开发的知识、信息随时可以获得	可能会在不同地点和时间上进行，所以需要保留开发信息以保证回归测试的正确进行
完成时间	所需时间长	只测试软件的一部分，所需时间短
执行频率	高频率的活动	由系统被修改而触发的周期性活动

1.2.4　回归测试过程

回归测试作为软件生命周期的一个组成部分，在整个软件测试过程中占有很大的工作量比重，软件开发的各个阶段都会进行多次回归测试。经研究表明，回归测试所花费的费用占系统维护成本的三分之一，因此回归测试的效率和有效性是重点考虑的问题。为了保证回归测试效率及有效性，通常情况下，项目测试组在软件开发过程中会将所开发的测试用例保存到“测试用例库”中，并对其进行维护和管理。同时，选择合适的回归测试策略也是回归测试效率和有效性的重要保证。