高等职业院校规划教材·软件技术系列

软件测试理论与实践

毛志雄 主编
余小斐 张 华 熊登峰 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等职业院校规划教材·软件技术系列

软件测试理论与实践

毛志雄　主编

余小斐　张华　熊登峰　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书主要介绍软件测试实际工作中的技能及其相关的实用软件测试理论。通过丰富的项目实践及培训经验，结合大量实际案例讲解软件测试技术和软件测试工具的应用，将最实用的技能及知识传授给读者，使读者尽快上手，少走弯路。书中每一个知识点都结合相关实例来讲解，完全按照实际工作当中软件测试的流程、方法来组织编写教材。

本书鲜明地突出了软件测试的实践性，共分为三部分。第一部分主要引导读者认识、了解软件测试这门学科以及这项工作的重要性，掌握一些必备的软件测试知识；第二部分着重讲解软件测试的基本理论和技能，包括黑盒测试、白盒测试、测试流程及管理、自动化测试等；第三部分给出了详细的测试用例设计方法、常用的自动化测试工具及一个详尽的软件测试项目案例，使读者真正掌握软件测试工作的实际技能。

本书适合高职高专院校的软件测试专业、软件技术专业及计算机相关专业使用，也可作为各类职业教育机构的软件测试培训教材，同时还可供从事软件开发和软件测试的专业技术及管理人员参阅。

图书在版编目（CIP）数据

软件测试理论与实践 / 毛志雄主编. —北京：中国铁道出版社，2008.8
高等职业院校规划教材. 软件技术系列
ISBN 978-7-113-08961-0

Ⅰ. 软… Ⅱ. 毛… Ⅲ. 软件—测试—高等学校：技术学校—教材 Ⅳ. TP311.5

中国版本图书馆 CIP 数据核字（2008）第 126964 号

书　　名：软件测试理论与实践
作　　者：毛志雄　主编

策划编辑：严晓舟　秦绪好
责任编辑：王占清　　**编辑部电话**：（010）63583215
编辑助理：王春霞　张国成
封面设计：付　巍　　**封面制作**：白　雪
责任印制：李　佳

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号　　邮政编码：100054）
印　　刷：三河市宏达印刷有限公司
版　　次：2008 年 8 月第 1 版　　2008 年 8 月第 1 次印刷
开　　本：787mm×1092mm　1/16　　**印张**：14.5　　**字数**：333 千
印　　数：4 000 册
书　　号：ISBN 978-7-113-08961-0/TP・2917
定　　价：23.00 元

前言

FOREWORD

软件测试是一门学科同时也是一门艺术，它要求从业人员不仅具有丰富的软件理论知识，也要有从各个角度衡量评价软件质量的能力，这样才能客观地欣赏软件的优点并找出软件的缺陷。如果说软件行业是朝阳产业，软件测试则是软件行业的朝阳产业。

软件的复杂性和规模正在不断地增加，软件工程和软件质量越来越成为软件产业发展的瓶颈，对软件测试的忽略和运用不够是最主要的原因。随着人们对软件本质的进一步认识，软件测试对于软件质量的保证已被软件企业高度重视，软件测试在软件开发中的作用也彰显出来，软件测试的地位得到空前提高。

本书为适应高职高专院校软件专业中软件测试课程教学的需要，理论联系实际，为培养既有深厚理论知识又有丰富实践能力的高技能人才而编写。本书内容丰富，涵盖了软件测试的各项基本技能和知识。在编写教材时，注意教材内容的先进性，将软件测试的新概念、新技术、新方法编入教材中。在内容的安排上，注意由易到难，深入浅出，多种实例，使学生能系统地掌握软件测试的理论知识和实践技能。

全书分 3 个部分共 13 章，第一部分包括第 1 章和第 2 章。第 1 章是对软件测试的概述，让读者从整体上了解软件测试这门学科，从而把握本书的脉络。第 2 章从软件工程的角度以及软件开发的过程来描述软件测试的阶段，认识软件测试贯穿整个软件生命周期。

第二部分包括第 3～9 章，共 7 章，着重讲解软件测试基本理论和技能，详细地介绍了白盒测试、黑盒测试、软件缺陷报告、软件测试流程、软件测试项目管理、自动化测试及其他专项测试理论和技术。

第三部分包括第 10～13 章，共 4 章，分别介绍测试用例设计、目前流行的自动化测试工具，具有很强的实用价值。最后一章用一个详尽的软件测试工程实例，全面阐述了在实际工作中软件测试的过程、理论运用、用例设计、文档编写和测试管理，具有很高的参考价值。

由于编者水平所限，加上时间仓促，书中错误和不妥之处在所难免，恳请读者批评指正，提出宝贵意见和建议。

编 者

2008 年 7 月

Learn more about it!

笔记栏

CONTENTS

目 录

第一部分 软件测试认知篇

第二部分 软件测试基本技能篇

第三部分 软件测试实践篇

第 1 章 软件测试概述

本章将学习软件测试所涉及的各个方面的基础知识，通过对本章的学习，读者能正确理解软件测试背景、软件缺陷和故障等概念以及软件测试的定义，概要了解软件测试的基本方法与过程，认识软件开发与软件测试相辅相成的关系。

学习本章的目的是使读者能对软件测试建立起概要性、框架性的整体认识，并为后续章节测试技术的学习打好基础。

1.1 什么是软件

什么是软件？这个问题虽简单但却不太好回答。现在被人们普遍认可的软件定义为：软件（software）是计算机中与硬件（hardware）相结合的一部分，包括程序（program）和文档（document）。

可以简单地用下式来表示：

软件=程序+文档（将复杂的概念简化为公式，可以方便记忆）

其中，“程序”指的是能够实现某种功能的指令的集合，如 C 语言程序、Java 程序、Visual Basic 程序等；“文档”指的是软件在开发、使用和维护过程中产生的图文集合，如系统需求规格说明书、用户手册、readme，甚至是一些软件宣传材料、包装文字和图形等。

从上面的软件定义中可以了解到，软件测试绝不等同于程序测试，文档测试也是软件测试的一个重要组成部分，这也是初学者甚至是测试工程师容易忽略的地方。图 1-1 所示为测试的分类。

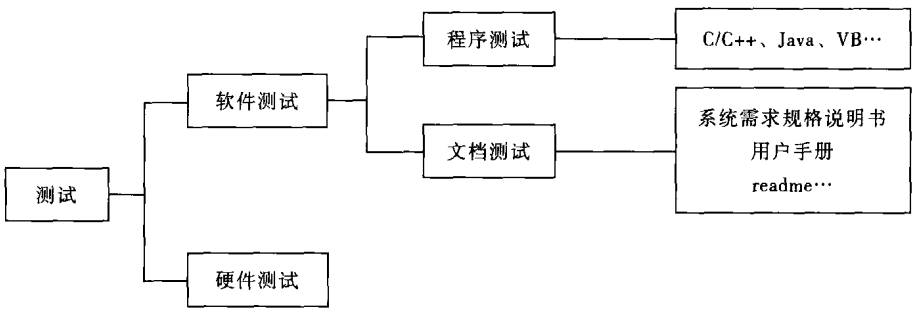

图 1-1 测试的分类

程序测试主要包括程序逻辑功能、界面、性能、易用性、兼容性、安装等的测试，文档测试主要包括文档内容和截图的检验，排版风格的检查，错别字的校验等。

1.2 软件的分类

软件有哪些类型？这个问题是我们从事软件测试之前应该弄清楚的问题。软件可从不同的角度分为多种类型。

1.2.1 按照功能划分

按照功能划分，软件可分为以下几种：

① 系统软件：能够直接操作底层的硬件并为上层软件提供支撑的软件，如操作系统软件、各种硬件驱动程序等。这类软件需要结合底层的硬件加以测试。

② 应用软件：能够为用户提供某种特定的应用服务的软件，如 Office、金山词霸、QQ 等。这类软件也是今后测试的重点。

1.2.2 按照技术架构划分

按照技术架构划分，软件可分为以下几种：

① 单机版软件：直接在单个计算机上安装并运行的软件，如 Office、画图工具等。这类软件的测试相对比较简单，不需要考虑网络传输。

② C/S 结构软件：C 指的是客户端（client），S 指的是服务器端（server），这种软件是基于局域网或互联网的，需要有一台服务器来安装服务器端软件，每台客户端都需要安装客户端软件。大家经常用的 QQ、MSN 以及各种网络游戏就属于 C/S 结构软件，如果想使用 QQ 聊天工具，就必须下载并安装一个 QQ 客户端，随后通过互联网连接深圳腾讯公司的 QQ 服务器来登录并使用 QQ。C/S 结构的软件过去比较流行，但由于不便于升级和维护（升级时需要重新安装每一个客户端），现在已逐渐被 B/S 结构的软件所取代。

③ B/S 结构软件：B 指的是浏览器（browser），S 指的是服务器（server），这种软件同样是基于局域网或互联网的，它与 C/S 结构软件的区别就在于不需要安装客户端，只需要有 IE 等浏览器即可。搜狐、新浪等门户网站以及 163 邮箱都属于 B/S 结构软件，用户只需要在 IE 浏览器的地址栏键入服务器的域名（如 http://mail.163.com）就可以直接访问并使用服务器端的程序了。B/S 结构的软件是现在软件的主流，与 C/S 结构的软件相比，便于升级和维护（升级时只需要升级服务器即可），也是今后测试的重点。

1.2.3 按照用户划分

按照用户划分，软件可分为以下几种：

① 产品软件：目标用户是大众用户，而不是某一特殊群体，如微软公司的 Office 软件、瑞星公司的瑞星杀毒软件等，它们共同的特点是针对的是千家万户的大众消费者，而不是固定的某一类用户。测试这类软件相对来说比较麻烦，因为最终用户使用的计算机系统千差万别（硬件有品牌机、兼容机；操作系统有 Windows 98、Windows 2000、Linux 等），需要考虑硬件和软件的兼容性测试。

② 项目软件：目标用户是具体的用户，而不是针对千家万户，如为北京市政府开发的办公自动化系统，它的特定用户就是北京市政府的领导和工作人员。国内 80 %以上的软件都属于项目软件，因此该类软件也是今后测试的重点。

1.2.4　按照开发的规模划分

图 1-2 是按照软件开发的参与人数以及开发时间来划分的。到此，可能有的读者会有一些迷惑，这么多的软件分类，它们之间到底是什么关系？其实它们只是分类的角度不同而已，一种软件按照不同的角度，可以有多种分法，如 QQ，它既是应用软件，也是 C/S 结构软件，又是产品软件；又如，Windows 98，它既是系统软件，也是单机版软件，又是产品软件。

类别	参与人数	开发时间
小型	10 人以下	1～4 个月
中型	10～100 人	1 年以下
大型	100 人以上	1 年以上

图 1-2　软件规模

1.3　什么是 Bug

Bug，在英语里是“小虫子”的意思，现在泛指计算机中硬件或软件的错误。硬件的出错有两个原因，一是设计错误，二是硬件老化失效。软件的错误全部是厂家设计错误。用户可能会执行不正确的操作，如本来是做加法运算但按了减法键，这样用户会得到一个不正确的结果，但不会引起 Bug 发作。软件厂商在设计产品时的一个基本要求，就是不允许用户进行非法的操作。允许用户做的都是合法操作。用户根本没有办法知道厂家心里是怎么想的，哪些操作是非法的。

从计算机诞生之日起，就有了 Bug。第一个有记载的 Bug 是美国海军的编程员，编译器的发明者格蕾斯·哈珀（Grace Hopper）发现的。哈珀后来成了美国海军的一个将军，主持了著名计算机语言 Cobol 的开发。

1945 年 9 月 9 日，下午 3 点，哈珀中尉正领着他的小组构建一个称为“马克二型”的计算机。这还不是一个完全的电子计算机，它使用大量继电器，是一种电子机械装置。第二次世界大战还没有结束，哈珀的小组夜以继日地工作。他们使用的机房是一间第一次世界大战时建造的老建筑，室内温度很高，没有空调，所有窗户都敞开散热。

突然，“马克二型”死机了，技术人员试了很多办法，最后定位到第 70 号继电器出错。哈珀观察这个出错的继电器，发现一只飞蛾躺在中间，已经被继电器打死。他小心地用镊子将蛾子夹出来，用透明胶布粘到“事件记录本”中，并注明“第一个发现虫子的实例”。

从此以后，人们将计算机错误地称为 Bug，而把寻找错误的工作称为 debug。

哈珀的事件记录本，连同那个飞蛾，现在陈列在美国历史博物馆中。

目前，计算机软件 Bug 之多是令人难以置信的。据计算机业界媒体报道，Windows 98 操作系统改正了 Windows 95 里面 5 000 多个 Bug。也就是说，Windows 95 软件推向市场时，每套里都含有 5 000 多个 Bug。

计算机软件含有很多 Bug 有一个技术原因，就是软件越来越庞大、复杂。在任何复杂的大型软件系统中，错误是难以避免的，大型软件尤其难以按时按预算完成。1995 年，国外一个大规模研究机构调查了 17 万个软件开发项目（总投资达 2 500 亿美元），结果发现，只有 6%按时按预算完成，31%的项目被中途取消，其余 63%的项目最终完成，但都超出了预算和进度。这些项目中，一大半项目的实际花费超出预算达 189%。

计算机程序是由语句组成的。据报道，Windows 95 含有 1 500 万行代码。假设每行代码包含一个语句，那么 Windows 95 中潜在的 Bug 就有 200 多万个。在出厂之前，微软做了大量测试。根据琼斯规则 5，需要做 18 次测试才能把 Bug 数降低到 5 000 个。假设测试一次耗时一个月（实际上常常不止一个月），就需要一年半的时间。如果要把 Bug 个数降到 1，总共需要做 42 次测试，需三年半还多的时间。当然，Windows 95 建筑在 Windows 3.1 版多年的开发和使用基础上，并不是完全从头做起，用不了这么多时间，但不论怎么算，测试和纠正 Bug 的成本都很大。

现在读者可以试着自己为软件 Bug 下一个明确的定义。

有人说软件的 Bug 就是指程序运行时出现的故障，下面一起来分析一下这句话是否正确。

举两个例子来说明这句话是不全面的。大家知道文档是软件的一部分，那么文档中的错误算不算 Bug 呢？显然，如果只说 Bug 是程序的错误就片面了。还有，运行一个软件，没有产生任何故障，但是软件实现的功能并不是用户需要的功能，如用户需要该软件进行英译汉的工作，而实际上却是汉译英，这当然也是 Bug。

现在给软件 Bug 下一个相对来说比较确切的定义：软件的 Bug 指的是软件中（包括程序和文档）不符合用户需求的问题。这个定义是我们判断一个软件问题是否是 Bug 的唯一标准。

从上面这个定义出发，可以将常见的软件 Bug 分成 3 种类型：

① 完全没有实现的功能。如用户需要在软件中实现 A、B、C 共 3 个功能，而只实现了 A、B 两个功能，没有实现 C 功能，则可以将此看做一个 Bug。

② 基本实现了用户需要的功能，但是运行时会出现一些功能或性能上的问题。例如某些软件能够满足用户的需求，但是运行的时候经常报错或导致系统死机，就属于这一类；又如，某订票系统的航班查询时间，用户要求是在 5s 以内，而实际的查询时间平均要在 10s 左右，同样属于这一类，而且属于性能上的 Bug。

③ 实现了用户不需要的功能，即多余的功能。如用户需要在软件中实现 A、B、C 3 个功能，但开发者却实现了 A、B、C、D 4 个功能，则多余的功能 D 可以看做一个 Bug。有的读者可能有疑问，软件有一些附加的功能不是更好吗？算是 Bug 吗？通常，判断是否是 Bug 的唯一标准就是软件是否符合用户的需求，用户不需要的功能就是不符合用户需求的功能，当然也算是 Bug。

现在读者对 Bug 的含义应该有一个更加深刻的认识。在实际工作中，发现 Bug、记录 Bug、关闭 Bug 是测试工程师的主要工作，而在关于一个现象是否是 Bug 的问题上，往往也是开发人员和测试人员争论的焦点，这就需要我们根据 Bug 的定义以及具体的软件环境来判断一个问题是否是 Bug。

还有一个关于用户需求的问题，用户需求相对来说比较抽象，在一般的软件项目组里，用户需求一般体现在“系统需求规格说明书”中（specification，一般由系统分析员或项目经理根据用户需求撰写），它是测试时需要参考的重要文档。但是要记住，“系统需求规格说明书”绝对不等同于用户的需求，因为从真实的用户需求到形成“系统需求规格说明书”，中间会有信息传递的误差，可以用图 1-3 来表示。

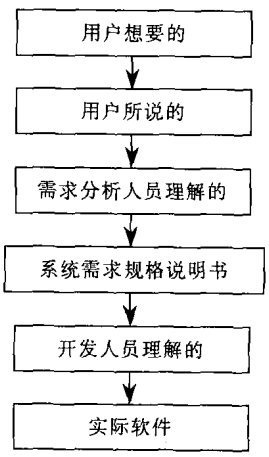

图 1-3 信息传递的误差

从图 1-3 可以看出，用户所说的不一定是用户实际想要的（用户需求），可能由于表达能力的限制，或是干脆自己也不清楚实际的需要，这种情况笔者经常遇到；需求分析人员所理解的也不一定是用户所说的，这里面可能会有理解的偏差和信息传递的误差；而需求分析人员所写的“系统需求规格说明书”不一定完全是他所理解的，这可能由书写

能力的限制等原因所致。最后，开发人员再根据自己对“系统需求规格说明书”的理解来开发软件。可想而知，实际的软件产品往往会与用户的真实需求有一定偏差。

所以说，无论是系统分析员编写的“系统需求规格说明书”，还是开发人员开发出来的实际软件，都不能完全代表用户的真实需求。建议大家在实际的工作中，尽可能多地争取与用户直接交流，参与系统的需求调研和评审，以获取用户的真实需求，这就需要测试人员具备较强的沟通能力。虽然现在测试人员很少参与需求调研，但这将会是今后一个发展趋势。

本节读者要把 Bug 的含义弄清楚，关于如何识别 Bug、提交 Bug 等后面将详细介绍。最后说明一点，本书如果没有特殊说明，则缺陷等同于 Bug。

1.4　什么是软件测试

测试的英文拼写为 TEST，测试包括硬件测试和软件测试，本书中如无特殊说明，特指软件测试，即 Software Testing。

据牛津英语大辞典记载，TEST 一词来源于拉丁语 TESTUM，原意是罗马人使用的一种陶罐（见图 1-4），在当时用它来评估像稀有金属矿石这样的材料的质量。从中我们可以看出，测试和产品质量的联系是很紧密的。

图 1-4　TESTUM（罗马人使用的一种陶罐）

软件测试的概念有早期定义和标准定义之分，下面逐一进行介绍。

1.4.1　早期定义

在 1979 年出版的一本经典著作《软件测试艺术》（*The Art of Software Testing*）中，Glemford J.Myers 曾经对软件测试下过如下定义：软件测试就是为了发现错误而执行程序或系统的过程。

下面通过两个例子来说明该定义的不完善之处：

① 测试文档属于软件测试，但是它不一定需要执行程序。

② 按照用户的需求测试了实际的系统，却一个 Bug 也没有发现，程序员所作的测试工作有意义吗?当然有意义，因为至少证明了该系统基本符合用户的需求。

其实上面的定义是把软件测试的目的和手段弄混了，发现错误仅仅是软件测试的手段而已，是副产品，软件测试的最终目的是检验实际的软件系统是否符合用户的需求，所以不能为了发现错误而发现错误。

当然，这个定义放在当时的环境下，是能够说得通的，因为那时的用户需求、质量保障等概念比较模糊，测试也仅仅是编码后的一个阶段，测试的主要工作也是用来发现错误的。

1.4.2　标准定义

软件测试的标准定义是：使用人工或自动手段来运行或测试某个系统的过程。其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别（1983 年，IEEE 软件工程标准术语）。

这个定义相对来说比较完善，它指出了软件测试的最终目的是检验预期结果（用户需求）和实际结果之间的差别。当然它也有不完善之处，如定义中说“运行或测试某个系统”，其实，软件测试并不一定要运行系统，测试的定义中也不应该再出现“测试”的字眼。