新世纪高职高专计算机软件技术专业规划教材

软件测试技术

贺平 编著

机械工业出版社
CHINA MACHINE PRESS

新世纪高职高专计算机软件技术专业规划教材

软件测试技术

贺　平　编著

机 械 工 业 出 版 社

本书是新世纪高职高专计算机软件技术专业规划教材之一。

本书主要介绍软件测试基础知识及相关的实用测试技术。内容包含两个部分：第一部分为基础概念、基础知识和基本测试技术，软件测试实质、软件测试策略、黑盒测试、白盒测试、集成测试、系统测试、确认测试、面向对象的测试，测试计划与测试文档；第二部分为软件自动化测试，介绍软件自动化测试基础知识，常用软件自动化测试工具的应用。

本书突出基本知识和基本概念的表述，注重技术方法的运用，力求内容全面，论述简明，深入浅出，通俗易懂，并注意将测试技术的应用与软件开发过程密切结合起来，使读者能较快地学习掌握当前的软件测试技术和实际运用。

本书适用于高职高专院校、软件职业技术学院软件测试课程使用，也可供从事软件开发、软件测试工作的管理与技术人员参考。

图书在版编目（CIP）数据

软件测试技术 / 贺平编著.—北京：机械工业出版社，2004.9

新世纪高职高专计算机软件技术专业规划教材

ISBN 7-111-15184-4

Ⅰ.软… Ⅱ.贺… 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（2004）第 087951 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：孙熹峻

封面设计：解 辰 责任印制：石 冉

保定市印刷厂印刷·新华书店北京发行所发行

2004 年 9 月第 1 版第 1 次印刷

1000mm×1400mm B5·8.25 印张·319 千字

定价：22.00 元

新世纪高职高专
计算机软件技术专业规划教材编审委员会

编写说明

党的十六大提出要走新型工业化道路，坚持以信息化带动工业化，以工业化促进信息化，加快发展现代服务业，全面建设小康社会。在推进国民经济信息化中，计算机应用、网络、软件专业人才的需求每年将在几十万人，为此教育部等六部门联合启动“制造业和现代服务业技能型紧缺人才培养培训工程”，同时教育部制定下发了“关于批准高等学校试办示范性软件职业技术学院的通知”，将计算机应用、网络及软件技术人才培养列入优先、快速和重点发展的地位。探索新的软件人才培养培训模式，把提高培养者的职业能力放在突出重要的位置，以应用为目的，构建就业导向的课程体系，坚持能力本位的课程设计原则，组织开发和编写具有鲜明特色的教材，是实施“工程”的目标任务之一。

根据上述高等职业教育的方针和软件人才培养的目标，本套教材编审委员会与机械工业出版社积极组织 IT 办学水平较高、教学改革成效显著的高职院校的计算机应用、网络及软件技术专业方面的学科带头人和教学骨干，开展产业人才需求调研、研讨人才培养模式、构建课程体系与教材开发等一系列工作。

在课程体系的构建中，注重对当前产业应用的主流 IT 技术清楚的认识，对 IT 企业人才需求全面的了解以及对 IT 技术发展的透彻理解和预见性的把握。同时在教材中突出以实践为主的原则，通过理论讲授、上机练习、案例教学、实际项目演练及企业实习等多种形式的教学内容介绍，强化技能训练，达到职业及专业能力培养。

本套教材体现了教学内容紧密结合专业核心能力对理论知识的要求，形成了有技术应用特点的理论知识体系，构成技术运用理论基础，满足了培养对象的需求。同时，注重融入信息技术的最新发展，更新内容，介绍新知识、新技术、新流程和新方法，把握主流技术和成熟技术的运用，实现专业教学基础性与先进性的统一。

本套教材还具有连贯性和递进性的特点，在实验、实训、实习、项目训练、工程训练的内容安排上力求具有新的特色，能反映专业岗位的工作需求，并成为软件人才成长的一套科学性、系统性、实用性较好的软件技术教育培训教材。

新世纪高职高专计算机软件技术专业规划教材编审委员会

目　录

第1章　软件测试概述

学习目标

1）正确理解软件测试的背景，软件缺陷和故障的概念。
2）正确理解软件测试的定义、目的和原则。
3）熟悉软件开发过程与软件测试的关系。
4）正确理解软件质量的概念及质量的管理。
5）了解 ISO9000 和 CMM 模型。
6）了解软件测试职业与素质的要求。

1.1　软件测试背景

随着计算机技术的迅速发展和越来越广泛深入地应用于国民经济及社会生活的各个方面，随着软件系统的规模和复杂性与日俱增，软件的生产成本、软件中存在的缺陷和故障造成的各类损失也大大增加，甚至会带来灾难性的后果。软件质量问题已成为所有使用软件和开发软件的人们关注的焦点。由于软件是人脑的高度智力化的体现这一特殊性，不同于其他科技和生产领域，因此软件与生俱来就有可能存在着缺陷。如何防止和减少这些可能存在的问题，答案是进行软件测试，测试是最有效的排除和防止软件缺陷与故障的手段，并由此促使了软件测试理论与技术实践的快速发展，新的测试理论、新的测试方法、新的测试技术手段不断涌现，与此同时，软件测试机构和组织迅速产生和发展，软件测试技术职业也同步完善和发展起来。

1.1.1　软件缺陷与定义

1．软件缺陷和软件故障的案例

如今，人类的生存和发展已经离不开各种各样的信息服务，为了提供这些信息，不仅需要计算机基础硬件，还需要各式各样的、功能各异的计算机软件，软件无处不在。然而，软件是由人编写的，因此软件就难免存在各种缺陷。下面是六例软件缺陷和故障的案例分析，借此说明软件缺陷和故障问题造成的损失和灾难。

案例 1　美国迪斯尼公司的狮子王游戏软件 bug（缺陷）

1994 年，美国迪斯尼公司发布面向少年儿童的多媒体游戏软件“狮子王动画故事书”。经过迪斯尼公司的大力促销活动，销售异常地火爆，使该软件游戏几乎

成为当年秋季全美少年儿童必买的游戏。但产品售后不久，该公司的客户支持部的电话就一直不断，愤怒的儿童家长和玩不成游戏的孩子们大量投诉该游戏软件的缺陷，一时间各种报纸和电视媒体也大量报道了这一游戏软件的各种问题。后经调查证实，造成这一严重问题的原因是迪斯尼公司没有对该游戏软件在已投入市场上使用的各种 PC 机型上进行正确的测试，也就是说游戏软件对硬件环境的兼容性没有得到保证。该游戏软件在开发该程序的程序员的机器硬件系统上工作是正常的，但在大众使用的常见系统中却存在不兼容问题。该软件故障使迪斯尼公司声誉大损，并为改正软件缺陷和故障付出了沉重的代价。

案例 2　美国航天局火星登陆事故

1999 年 12 月 3 日，美国航天局的火星极地登陆飞船在试图登陆火星表面时突然失踪。负责这一太空发展项目的错误修正委员会的专家们观测到并分析了这一故障，确定出现该故障的原因是由于某一数据位被意外地更改而造成灾难性的后果，并得出该问题应该在内部测试时就应予以解决的结论。

简要地说，火星登陆的过程是这样的：当飞船快要降落到火星表面时，它将打开着陆降落伞以减缓飞船的下落速度。在降落伞打开后的几秒钟内，飞船的三条支撑脚将迅速撑开，并在预定的地点着陆。在飞船距离火星 1 800 米时，飞船将丢弃降落伞，同时点燃登陆推进器，控制稳定飞船的下降速度，使其在余下的高度里缓慢降落到火星表面。

然而，美国航天局为了节省研制经费，简化了确定何时关闭登陆推进器的装置，为了替代其他太空船上通常使用的贵重的着陆雷达系统，设计师们在登陆飞船的支撑脚上安装了一个简易廉价的触电开关，并在计算机中设置一个数据位来控制关闭登陆推进器的燃料。很明确，飞船的支撑腿在没有着地之前，推进器引擎就会一直处于着火工作状态。

遗憾的是，在事后的分析测试中发现，当飞船的支撑脚迅速打开准备着陆时，机械振动很容易触发着地触电开关，导致设置了错误的数据位，关闭了登陆推进器的燃料。设想当飞船开始进入着陆动作时，由于机械震动的缘故，触发了着地触电开关，计算机极有可能关闭了推进器的燃料，也就是说登陆推进器提前停止了工作，使火星登陆飞船加速下坠 1 800 米之后直接冲向火星表面，撞成碎片。

这一事故的后果非常严重，损失巨大，然而起因却如此简单，是软件设计中的缺陷。事实是在飞船登陆飞行发射之前，飞船各部位的工作过程经过了多个小组的测试，其中一个小组测试飞船的支撑脚落地打开过程，另一个小组测试此后的着陆过程，前一个小组没有注意到着地数据位是否已经置位，因为这不属于他们负责的范围；而后一个小组总是在开始测试之前重置计算机，进行数据的初始化，清除数据位。双方的独立工作都很好，但从未在一起进行过集成测试，从而

导致了这一灾难性事故的发生。

案例 3　千年虫问题

这是一个非常著名的计算机软件缺陷问题，在 20 世纪末的最后几年中，全世界的各类计算机硬件系统、软件系统和应用系统都为“千年虫问题”而付出了巨大的代价。

20 世纪 70 年代，由于当时所使用的计算机内存空间很小，一位负责开发公司工资系统的程序员，被迫在程序设计时要考虑节省每一个字节，以减少对系统内存的占用。其中节约内存的措施之一就是把表示年份的 4 位数，例如 1973，缩减为 2 位，即 73。因为工资系统极度依赖数据处理，会有大量的数据占用内存空间，所以节约每一个字节的意义很大，该程序员的这一方法确实节省了可观的存储空间。他采用这一措施的出发点主要是认为只有到了 2000 年时程序在计算 00 或 01 这样的年份时才会出现问题，但在到达 2000 年时，程序早已不用或者修改升级了。1995 年，这位程序员退休了，但他所编制的程序仍在使用，没有谁会想到进入程序去检查 2000 年兼容的问题，更不用说去做修改了。计算机系统在处理 2000 年份问题（以及与此年份相关的其他问题）时软、硬件系统中存在的问题隐患被业界称为“千年虫问题”。

据不完全统计，从 1998 年初全球就开始进行“千年虫”问题的大检查，特别是金融、保险、军事、科学、商务等领域，花费了大量的人力、物力对现有的各种各样的程序进行检查、修改和更正，据有关资料统计，仅此项费用超过了数百亿美元。

案例 4　爱国者导弹防御系统炸死自家人

美国爱国者导弹防御系统首次应用于海湾战争中，以对抗伊拉克的飞毛腿导弹防御系统。尽管爱国者导弹防御系统在这次战争中屡建功勋，多次成功拦截飞毛腿导弹，但确实也有几次在对抗中失利，其中有一枚爱国者导弹在沙特阿拉伯的多哈炸死了 28 名美军士兵。事后，分析专家得出这一事故是爱国者导弹防御系统中一个软件系统的缺陷所致。一个很小的系统时钟错误积累起来就可能延时 14 个小时，造成跟踪系统失去准确度。在那次的多哈袭击战中，导弹系统的重要时刻被延时 100 多个小时，造成了这一悲剧。

案例 5　Windows 2000 中文输入法漏洞

在安装微软的 Windows 2000 简体中文版的过程中，在默认情况下会同时安装各种简体中文输入法。随后这些装入的输入法可以在 Windows 2000 系统用户登录界面中使用，以便用户能够使用基于字符的用户表示和密码登录系统。然而，在默认安装的情况下，Windows 2000 中的简体中文输入法不能正确检测当前的状态，导致了在系统登录界面中提供了不应有的功能，即出现了下面的问题：

在 Windows 2000 用户登录界面中，当用户输入用户名时，用“Ctrl+Shift”组合键将输入法切换到全拼输入法状态下，同时在登录界面的屏幕的左下角将会出现输入法状态条。用鼠标右键单击状态条并在出现的菜单中选择“帮助”项，再将鼠标移到“帮助”项上，在弹出的选择项里选择“输入法入门”，随后即弹出“输入法操作指南”帮助窗口。再用鼠标右键单击“选项”，并选择“跳至 URL”，此时将出现 Windows 2000 的系统安装路径并要求填入路径的空白栏。如果该操作系统安装在 C 盘上，在空白栏中填入“c:\windowsnt\system32”，并单击“确定”，在“输入法操作指南”右边的框里就会出现 c:\windowsnt\system32 目录下的内容了。也就是说这样的操作成功的绕过了身份的验证，顺利地进入了系统的 system32 目录，当然也就可以进行各种各样的操作了。

此软件缺陷被披露后，微软公司推出了该输入法的漏洞补丁，并在 Windows2000 Server Pack2 以后的补丁中都包含了对该漏洞的修补，但对于没有打补丁的用户来说，系统仍处于不安全的状态之中。

案例 6　金山词霸 bug

在国内，“金山词霸”是一个很著名的词典软件，应用范围极大，对使用中文操作的用户帮助很大，但它也存在不少 bug。如输入“cube”，词霸会在示例中显示 $3^3=9$ 的错误；又如，如果用鼠标取词“dynamically”（力学，动力学），词霸会出现其他不同的单词“dynamite n，炸药”的显示错误等等，不同的版本都存在这个问题。

诸如上述的软件错误或漏洞绝不仅仅是这几例，一些著名软件的缺陷、错误经常在互联网上被用户披露或指出，同时也不断有经过修改的软件版本在发布。

2．软件缺陷定义

从以上例子中可以看到软件发生错误时造成的灾难性危害或对用户的各种影响。

那么，这些事件的共同特点有哪些呢？首先，软件开发过程没有按照预期目标进行；其次，软件虽然都经过了测试，但并不能保证完全排除了存在（潜在的）的错误。对于软件测试来说，其任务就是要发现软件中所隐藏的错误，找出那些不明显的、小到难以察觉的、简单而细微的错误，这是对软件测试人员的最大挑战。

上述所有实例中的软件问题在软件工程或软件测试中都被称为软件缺陷。不管软件存在问题的规模和危害有多大，由于都会产生使用上的各种障碍，所以将它们统称为软件缺陷。

对于软件缺陷的精确定义，通常有下列 5 条：

1）软件未达到产品说明书中已经标明的功能。

2）软件出现了产品说明书中指明不会出现的错误。

3）软件未达到产品说明书中虽未指出但应当达到的目标。

4）软件功能超出了产品说明书中指明的范围。

5）软件测试人员认为软件难以理解、不易使用，或者最终用户认为该软件使用效果不好。

现在，以常用的计算器的软件来理解上述每条定义的规则。

计算器说明书一般声称该计算器将准确无误进行加、减、乘、除运算。如果测试人员或用户随意按下了“+”号键，结果没有任何反应，根据第一条规则，这是一个软件缺陷；如果得到错误答案，根据第一条规则，同样是软件缺陷。

假如计算器产品说明书指明计算器不会崩溃、死锁或者停止反应。而在用户狂敲键盘后，计算器停止接受输入，根据第二条规则，这也是一个软件缺陷。

如果在测试过程中发现，因为电池没电而导致计算不正确，但产品说明书未能指出在此情况下，应如何进行处理，根据第三条，也应算作软件缺陷。

如果进行测试，发现除了规定的加减乘除功能之外，还能够进行求平方根的运算，而这一功能并没有在说明书的功能中规定，根据第四条规定，也是软件缺陷。

第五条的规则说明了无论测试员或者是用户，若发现计算器某些地方不好用，比如按键太小、显示屏在亮光下无法看清等等，也都算做是软件缺陷。

1.1.2 软件缺陷产生的原因

软件测试是在软件投入运行之前，对软件需求分析、设计规格说明和编码实现的最终审定。那为什么会产生软件缺陷呢？经过研究发现，表现在程序中的故障，并不一定是由编码所引起的。大多数的软件缺陷并非来自编码过程中的错误，从小项目到大项目基本上都是如此，因为软件缺陷很可能是详细设计、概要设计阶段，甚至是需求分析阶段的问题所致，即使是针对源程序进行的测试所发现的故障的根源也可能存在于软件开发前期的各个阶段。大量的事实表明，导致软件缺陷的最大原因是软件产品说明书。

在多数情况下，软件产品说明书写得不清楚、不全面，在开发过程中经常被更改，或者开发小组的人员之间没有很好的进行交流与沟通，都会导致软件缺陷。因此，软件产品开发计划是非常重要的，如果计划没有做好，软件缺陷就会出现。

软件缺陷产生的第二大来源是设计方案，它是实施软件计划的关键环节。

图 1-1 是软件缺陷产生的原因分布图，其中表明占据大部分的是由于编写说明书（需求）的原因造成软件缺陷的。

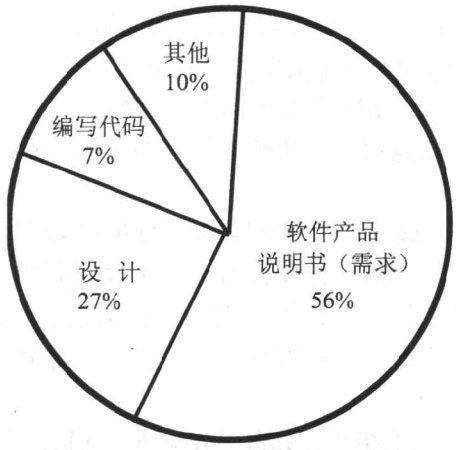


图 1-1 软件缺陷产生的原因分布

1.1.3 软件缺陷修复的代价

软件通常需要靠有计划、有条理的开发过程来建立。从计划、编制、测试、一直到交付用户公开使用的过程中，都有可能产生和发现缺陷。随着整个开发过程的时间推移，修复软件的费用呈几何级数增长。如果说在编写产品说明书这样的早期发现的软件缺陷，修正费用是按元计算，那么同样的软件缺陷在软件编制完成开始测试的时候才发现，修正费用将要上升十倍，如果软件缺陷是在发售后由用户发现则修正费用可能达到上百倍。图 1-2 是在不同阶段发现软件缺陷时修改费用增长示意图。

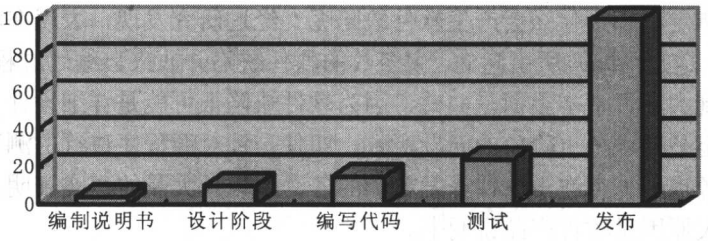


图 1-2 在不同阶段发现软件缺陷时修改费用示意图

这个过程说明了越是在软件开发过程的早期发现软件的缺陷，修正缺陷的费用就越低，反之代价是很大的。如前面案例“狮子王动画故事书”的软件缺陷的例子，根本的问题是软件无法在当时流行的家用 PC 机平台上运行。假如早在编写产品说明书时就考虑过几种较为流行的家用 PC 机平台，并在说明书上写明需要在某种机型的配置上进行设计和测试，所花费的人力和财力并不大，也不会等到产品发售、到达用户手中后出现大量问题而造成被动局面。在这一软件缺陷的

恶性事件中，迪斯尼公司最终为客户进行了产品退货、软件更换，同时，进行了新一轮的软件产品设计、调试、修改和测试，不仅花费了大量的费用，还损失了已有的良好声誉。

1.2 软件测试的定义、目的和原则

1.2.1 软件测试的定义

1．关于软件测试的常用术语

（1）测试

1）测试是一项活动，在这项活动中某个系统或组成的部分将在特定的条件下运行，结果将被观察和记录，并对系统或组成部分进行评价。测试活动有两个结果，找出缺陷或故障，显示软件执行正确。

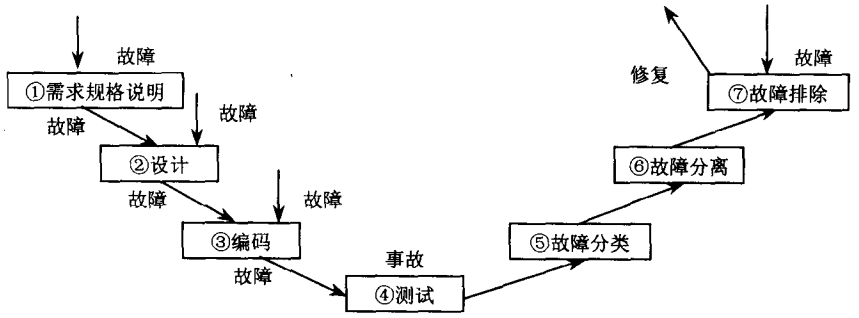
2）测试是一个或多个测试用例的集合。

（2）测试用例

1）测试用例是为特定的目的而设计的一组测试输入、执行条件和预期的结果。

2）测试用例是执行的最小实体。

（3）测试步骤　测试步骤则详细规定了如何设置、执行、评估特定的测试用例。图 1-3 是一个测试生命周期模型。



其中：①、②、③可能引入故障或导致其他阶段的故障，为第一阶段；

④ 测试发现事故，为第二阶段；

⑤、⑥、⑦ 清除故障，为第三阶段；

⑦ 故障排除的过程有可能使以前正确执行的程序出现错误，引入了新的故障。

图 1-3　一个测试生命周期

2．软件测试定义

软件测试就是在软件投入运行前，对软件需求分析、设计规格说明和编码的最终复查。它是软件质量保证的关键步骤。通常对软件测试的定义有两种描述：