普通高等教育“十一五”国家级规划教材

陈　明 编著

软件测试技术

21世纪计算机科学与技术实践型教程

丛书主编　陈明

清华大学出版社

内 容 简 介

本书是计算机软件测试课程教材，主要内容包括软件测试概述、软件测试方法、软件测试过程、面向对象测试、测试的设计与实现、Web测试、软件测试自动化、软件质量与质量保证、软件测试工具等内容。

本书可作为高等学校计算机科学与技术专业的软件测试课程教材，也可作为计算机软件开发人员的参考书。

图书在版编目（CIP）数据

软件测试技术/陈明编著．—北京：清华大学出版社，2011.2

（21世纪计算机科学与技术实践型教程）

ISBN 978-7-302-23780-8

Ⅰ．①软…　Ⅱ．①陈…　Ⅲ．①软件—测试—高等学校—教材　Ⅳ．①TP311.5

中国版本图书馆CIP数据核字(2010)第171220号

责任编辑：谢　琛　王冰飞

责任校对：李建庄

责任印制：王秀菊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座

http://www.tup.com.cn　　**邮　　编**：100084

社　总　机：010-62770175　　**邮　　购**：010-62786544

投稿与读者服务：010-62795954，jsjjc@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：185×260　　**印　　张**：12.75　　**字　　数**：312千字

版　　次：2011年2月第1版　　**印　　次**：2011年2月第1次印刷

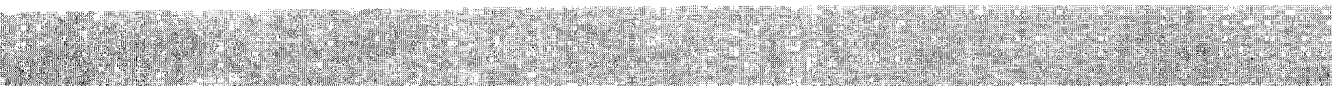
印　　数：1～4000

定　　价：25.00元

产品编号：039513-01

《21世纪计算机科学与技术实践型教程》

编辑委员会



前　言

计算机软件是逻辑产品。软件与硬件具有完全不同的特征。计算机软件现已成为一种新的驱动力，是进行决策的引擎，是现代工程研究和解决问题的基础。在各种类型的应用系统中无所不在，具有十分广泛的应用。

随着软件企业规模的扩大，复杂度的不断提高，软件测试难度也进一步加大，凸显了软件测试的重要性。软件测试是软件工程学科的重要分支，现已成为软件质量保证的关键技术之一。在软件开发过程中，软件测试是不可缺少的重要环节，软件测试工作直接决定了软件产品的质量。

软件测试工具是支持软件生存周期中某一阶段的测试任务实现而使用的计算机程序。软件测试环境是一组相关的软件测试工具的集合，将它们集成在一起支持软件测试。软件测试工具与环境是软件测试的重要组成部分，对于提高软件生产率，改进软件质量有越来越大的作用。

软件测试是异常活跃的技术，需要丰富的想象力。软件测试又是一个实践性极强的实用技术，在学习中，既要学习基本的理论知识，又要掌握必要的技能，也就是说，不仅要能掌握其理论原则与方法，更重要的是能熟练地进行应用。软件测试人才的需要日益增多，通过软件测试的理论学习与实践，可以培养学生掌握测试的基本内容和方法，并在软件开发的工作中得以贯彻，进而展现学科的力量。

在学习软件测试技术过程中，要注重技术的应用，通过大量的时间和思考，理解软件测试的思想和理念，并运用测试技术和技巧去解决问题。

全书分为 9 章，主要包括软件测试概述、软件测试方法、软件测试过程、面向对象测试、测试的设计与实现、Web 测试、软件测试自动化、软件质量与质量保证、软件测试工具等内容。

在内容选择上，注重先进与系统；在结构上，各章呈模块化。在描述中，面向实践，注重理论与实践的结合，有助于快速掌握软件测试必需的技术和方法，促进软件测试能力的培养。

由于作者水平有限，书中不足之处在所难免，敬请批评指正。

陈明

2010 年 12 月于北京

第1章 概 述

学习要点：

- ❖ 软件缺陷。
- ❖ 软件测试的定义。
- ❖ 软件测试的对象。
- ❖ 软件测试的目的。

计算机科学技术的飞速发展，促进了软件产品的广泛应用，不论是软件的生产者还是使用者，都在激烈的竞争中求生存，软件产品的质量已经成为关注的焦点。软件开发者为了占有市场，必须把产品质量作为企业的重要目标之一，进而才可以确保在激烈的竞争中获得胜利。为了保证软件产品的质量，软件测试成为必不可少的重要过程与手段。

在开发大型软件系统的过程中，面对极其错综复杂的问题，软件开发者的主观认识不可能完全符合客观事实，而且与工程密切相关的各类人员之间的沟通和配合也不可能完美无缺，因此，在软件生存周期的每个阶段不可避免地产生差错。尽管力求在每个阶段结束之前通过严格的技术审查，尽可能早地发现并纠正差错，但是，审查并不能发现所有错误，此外，在这些错误存在的过程中还可能引入新的错误。如果软件在投入实际运行之前，没有发现并纠正它存在的部分错误，那么这些错误将在运行过程中暴露出来，不仅要为改正这些错误而付出高昂的代价，而且很可能产生严重后果。

软件测试在软件生存周期中经历两个阶段。在编写出每个模块之后就对它做测试，称为单元测试，模块的编写者和测试者是同一个人，编码和单元测试在软件生存周期中属于同一个阶段。在这个阶段结束之后，对软件系统还应该进行各种综合测试，这是软件生存周期中的另一个独立的阶段，由专门的测试人员承担这项工作。

大量统计结果表明，软件测试的工作量占软件开发总工作量的40%以上，在特殊情况下，例如对关系到人的生命安全的软件要进行的测试所花费的成本，可能相当于软件工程其他开发步骤总成本的3～5倍。因此，必须重视软件测试，绝不要以为编写出程序之后软件开发工作就完成了，实际上，几乎还需要完成与开发工作同样多的工作量。

1.1 软件测试的发展

随着社会化生产，应运而生的测试技术涉及多方面。在许多领域，测试都是保证产品质量的关键。软件测试是软件工程的一部分重要内容。

随着计算机的产生与发展，软件开发和软件测试相继出现。由于早期的计算机性能比较差，软件的可编程范围也比较狭窄，在这一阶段并没有系统的软件测试，更多的是一种调试性测试，测试主要是为了证明系统的可运行性。

20 世纪 50 年代后期到 20 世纪 60 年代，许多高级语言相继诞生并且得到了广泛的应用，测试的对象逐渐转入到用高级语言书写的系统。但是，由于受到硬件系统发展瓶颈的限制，软件测试位于次要地位，软件的正确性和可用性主要由编程人员的水平所决定。因此，软件测试理论和方法的发展缓慢。

20 世纪 70 年代以后，随着计算机处理速度的提高，软件在整个系统中的重要性变得越来越重要。一方面在这个阶段，软件的规模越来越大，可视化的编程环境、日益完善的软件分析设计方法以及新的软件开发过程模型的出现使得大型软件的开发成为可能；另一方面，由于软件规模和复杂性的迅速增加，软件面临着巨大的危机，软件测试得以重视并提到日程。

20 世纪 70 年代中期，软件测试技术的研究达到高潮。J. B. Goodenough 和 S. L. Gerhart 首先提出了软件测试的理论，从而把软件测试这一实践性很强的学科提高到理论的高度。1982 年 6 月在美国北卡罗来纳大学召开了首次软件测试的技术会议，讨论了软件测试问题，这次会议是软件测试技术发展中的一个重要里程碑。

软件产业的发展，对软件的成本、进度和质量都提出了更高的要求，对软件质量的控制已不再是传统意义上的软件测试。传统的测试一般在软件开发后期才介入，然而，大量研究结果表明设计活动引入的错误占软件开发过程中出现的所有错误的 50%～65%。因此，测试就已经不再是一个编码后才进行的活动，而是一个基于软件开发整个生存周期的质量控制活动。

目前，在软件测试理论、测试方法、测试过程和测试工具等方面的研究取得了大量的进展。这不仅使得软件的质量有了基本的保证，也使得软件测试的工作量占到了软件开发总工作量的 40%以上，软件测试的地位上升到前所未有的高度。

1.2 软件错误与软件缺陷

1.2.1 软件错误与缺陷的概念

1. 软件错误

在编写代码时有可能会出现错误，把这种错误叫做 Bug。错误在整个软件开发周期中很可能扩散，在需求阶段发生的错误，在设计期间有可能被放大，在编写代码时还会进一步扩大。

2. 软件缺陷

缺陷是错误的结果。更精确地说，缺陷是错误的表现，缺陷很难捕获。当设计人员出现遗漏错误时，所导致的缺陷会是遗漏本来应该在表现中提供的内容。这种情况表明需要对定义做进一步的细化。把缺陷分为错误缺陷和遗漏缺陷。如果把某些信息输入到不

正确的表示中，就是错误缺陷；如果在设计过程中没有输入某些正确且必要的信息，就是遗漏缺陷。在这两类缺陷中，后者更难检测和解决。

1.2.2 软件错误类型及出现的原因

1. 软件错误类型

根据软件错误的性质不同，可以把软件错误分为下述几种类型。

1）需求错误

软件需求指定的不合理或不正确；需求不完全；需求中含有逻辑错误；需求分析的文档有误，等等。

2）功能与性能错误

功能或性能存在错误，或是遗漏了某些功能，或是规定了某些冗余的功能；为用户提供的信息有错，或信息不确切；对异常情况处理有误，等等。

3）软件结构错误

程序控制流或控制顺序有误；处理过程有误，等等。

4）数据错误

数据定义或数据结构有误；数据存取或数据操作有误，等等。例如：动态数据与静态数据混淆、参数与控制数据混淆等。

5）实现和编码错误

编码错误包括语法错误、数据名错误、局部变量与全局变量混淆或者程序逻辑有误等。

6）集成错误

软件的内部接口、外部接口有误；软件各相关部分在时间配合、数据吞吐量等方面不协调，等等。

7）系统结构错误

操作系统调用错误或使用错误、恢复错误、诊断错误、分割及覆盖错误以及引用环境的错误等。

8）测试定义与测试执行错误

测试的错误包括测试方案设计与测试实施的错误、测试文档的问题、测试用例不够充分等。普遍出现的软件结构错误、数据错误和功能与性能错误特别受到重视。

2. 出现错误的原因

软件出现错误的原因是多方面的，归纳起来主要有如下几点。

（1）交流不够、交流上有误解或者根本没有进行交流。在不清晰应该做什么或不应该做什么的情况下进行了应用开发。

（2）软件复杂性。图形用户界面（GUI，Graphic User Interface），客户/服务器结构，分布式应用，数据通信，超大型关系型数据库以及庞大的系统规模，使得软件复杂性呈指数增长。

(3) 程序设计错误。在软件设计阶段出现的错误,主要包括概要设计、详细设计和编码步骤出现了错误。

(4) 需求不断变化。需求变化的后果可能是造成系统重新设计、项目日程的重新安排、已经完成的工作可能要部分重做甚至完全抛弃等。如果有许多小的改变或者一次大的变化,项目各部分之间已知或未知的关系相互影响进而导致更多问题的出现,还可能影响工程参与者的积极性。

(5) 时间压力。软件项目的日程表很难做到准确,很多时候需要预计和猜测。当最终期限到来之际,由于时间紧迫,容易出现错误。

(6) 代码文档不完全。在一些团队中,不鼓励其程序员为代码编写文档,也不鼓励程序员将代码写得清晰和容易理解,相反认为少写文档可以更快地进行编码,无法理解的代码更易于工作的保密,显然,这是一种错误的认识。

(7) 软件开发工具。当软件产品的开发依赖于某些工具时,那么这些工具本身隐藏的问题可能会导致产品的错误。因此,应该选择比较成熟的开发工具,而不是追求最先进的开发工具。

1.2.3 软件缺陷的主要特征

软件错误有多种类型,在一些关键系统中,出现软件错误时,其后果是灾难性的。而在非关键性系统中,出现错误的后果可能并不像前一种情况那样明显,但难以观察。在通常情况下,利用软件缺陷描述软件错误。软件缺陷的主要特征如下。

(1) 软件未达到软件产品需求说明书指明的要求。

(2) 软件出现了软件产品需求说明书中指明不应出现的错误。

(3) 软件功能超出软件产品需求说明书指明的范围。

(4) 软件未达到软件产品需求说明书未指明但应达到的要求。

(5) 软件测试人员认为难以理解、不易使用、运行速度慢或最终用户认为不好。

考虑到设计等方面的因素,软件缺陷还包括软件设计不符合规范,未能在特定的条件下(资金、范围等)达到最佳等。但是,更多的是把软件缺陷看成软件运行时出现的各种问题。

统计结果表明:大多数软件缺陷并非源自编码错误,导致软件缺陷的最大原因是需求分析错误,其次是设计错误,还有编码错误和测试错误等。

1.3 软件测试的定义

软件测试的定义是:软件测试是为了发现错误而执行程序的过程。这个定义明确指出寻找错误是测试的目的。

软件测试是软件工程过程的一个重要阶段,在软件投入运行前,对软件需求分析、设计和编码各阶段产品的最终检查,是为了保证软件开发产品的正确性、完全性和一致性,从而进行检测错误以及修正错误的过程。软件开发的目的是开发出满足用户需求的高质

量、高性能的软件产品，而软件测试以检查软件产品内容和功能特性为核心，是软件质量保证的关键步骤，也是成功实现软件开发目标的重要保障。

从用户的角度来看，普遍希望通过软件测试找出软件中隐藏的错误，所以软件测试应该是为了发现错误而执行程序的过程。软件测试应该根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例（即输入数据及其预期的输出结果），并利用这些测试用例去运行程序，以发现程序中隐藏的错误。

软件测试的主要作用如下。

(1) 测试是执行一个系统或者程序的操作。

(2) 测试是带着发现问题和错误的意图来分析和执行程序。

(3) 测试结果可以检验程序的功能和质量。

(4) 测试可以评估项目产品是否获得预期目标和可以被客户接受的结果。

(5) 测试不仅包括执行代码，还包括对需求等编码以外的测试。

1.4 软件测试的对象

软件开发期间任何一个环节发生了问题都可以在软件测试中表现出来。软件测试应该贯穿软件开发期间。因此需求分析、概要设计、详细设计以及程序编码等各阶段所得到的文档，包括需求规格说明、概要设计规格说明、详细设计规格说明以及源程序，都是软件测试的对象。

1.5 软件测试的目的

软件测试的目的就是找出被测试软件所有存在的错误，但实际上，测试人员不可能发现所有的错误。一个不成功的测试是没有找到错误的测试，成功的测试是花费最少的时间和人力找出软件中潜在的各种错误。

如果根据上述目标成功构造了测试，则能够揭示软件中存在的错误。测试能证实软件根据需求所具有的功能和性能，此外，在构造测试方案的过程中，收集的数据可以为软件可靠性以及软件的整体质量提供一些比较重要的信息。但是，测试无法说明错误不存在，只能揭示目前的软件的状态良好。

软件测试不以发现错误为唯一目的，查不出错误的测试并非没有价值。通过分析错误产生的原因和错误的分布特征，可以帮助发现当前所采用的软件过程中的缺陷并加以改进。同时，这种分析也能帮助设计出有针对性的检测方法，改善测试的有效性。没有发现软件中错误的测试也是有价值的，因为整个测试过程本身就是评定测试软件质量的一种方法。如果在运行多次后或者重新构建一套测试软件后而仍未发现软件错误，这样可以得出这样的结论：被测试软件已经比较完美。因为存在不同的针对性，所以软件测试也就存在多种目的，其中最重要的三条如下。

(1) 证明测试人员所做的是客户所需的。

(2) 确保编程人员正确理解设计的意图。

(3) 通过回归测试来保证目前运行的程序在将来仍然可以正常工作。

测试目的决定了测试方案的设计。如果为了表明程序是正确的而进行测试,就很可能设计一些不易暴露错误的测试方案;相反,如果测试是为了发现程序中存在的错误,就会设计出最能暴露错误的测试方案。

1.6 软件测试的原则

经过理论分析和工作实践,总结如下一些软件测试原则。

1. 尽早不断测试的原则

应当尽早不断地进行软件测试。据统计,约 60%的错误来自设计以前,并且修正一个软件错误所需的费用将随着软件生存周期的进展而上升。错误发现得越早,修正它所需的费用就越少。

2. IPO 原则

测试用例由测试输入数据和与之对应的预期输出结果这两部分组成。

3. 独立测试原则

独立测试原则是指软件测试工作由在经济上和管理上独立于开发机构的组织进行。程序员应避免检查自己的程序,程序设计机构也不应测试自己开发的程序。软件开发者难以客观、有效地测试自己的软件,而找出那些因为对需求的误解而产生的错误就更加困难。采用独立测试原则的优点如下所述。

(1) 客观性:经济上的独立性使其工作有更充分的条件按测试要求去完成。

(2) 专业性:软件测试需要有专业队伍加以研究,并进行工程实践。专业化分工是提高测试水平、保证测试质量、充分发挥测试效率的必然途径。

(3) 权威性:由于专业优势,独立测试工作形成的测试结果更具有信服力和权威性。

4. 合法和非合法原则

在设计时,测试用例应当包括合法的输入条件和不合法的输入条件。

5. 错误群集原则

软件错误呈现群集现象。经验表明,某程序段剩余的错误数目与该程序段中已发现的错误数目成正比,所以应该对错误群集的程序段进行重点测试。

6. 严格性原则

严格执行测试计划,排除测试的随意性。

测试计划应包括所测软件的功能,输入和输出,测试内容,各项测试的进度安排,资源要求,测试资料,测试工具,测试用例的选择,测试的控制方法和过程,系统的组装方式,跟踪规则,调试规则,回归测试的规定以及评价标准等。