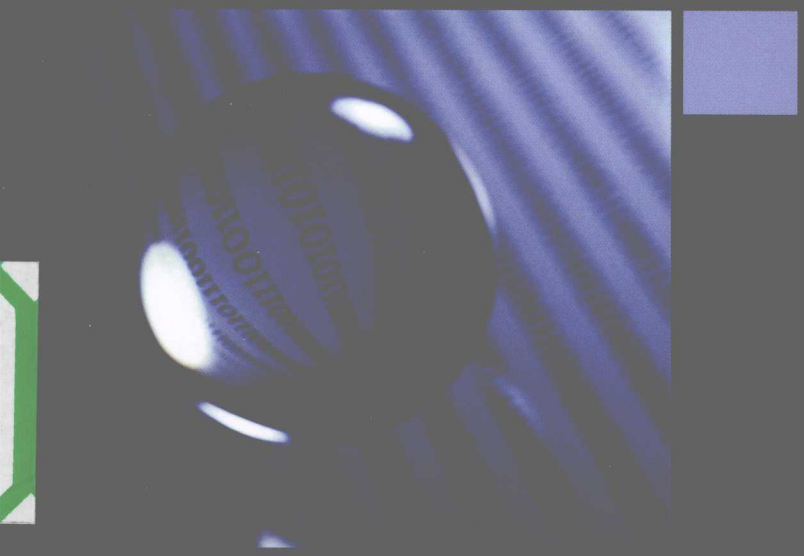
软件研发精品译丛

Software Testing
Concepts and Practices

软件测试：概念与实践

K.Mustafa & R.A.Khan 著
董威 译



科学出版社
www.sciencep.com

软件研发精品译丛

软件测试：概念与实践

Software Testing: Concepts and Practices

K. Mustafa & R. A. Khan 著
董 威 译

科学出版社
北 京

图字:01-2008-3376 号

内 容 简 介

本书系统地讲述软件测试的基本概念、理论和方法,及其在工程实践中的应用。本书中,软件测试的概念作为软件开发过程中每个阶段的一个有机组成部分进行讲述,而不是像传统方式那样把软件测试作为独立的、位于软件实现之后的一项活动。书中每一章的开始都给出一组预期要达到的目标,以方便读者阅读;每一章的结尾都给出相关的参考文献,以方便读者进行深入学习。

本书适用于高等院校的计算机专业本科及硕士生阅读,可作为软件测试课程教材。书中阐述了软件测试研究和实践领域的实用内容,可供软件测试领域的研究者和工程实践人员阅读参考。

图书在版编目(CIP)数据

软件测试:概念与实践/穆斯塔法(Mustafa,K.),库翰(Khan,R. A.)著;董威译. —北京:科学出版社,2009

(软件研发精品译丛)

ISBN 978-7-03-023095-9

Ⅰ.软… Ⅱ.①穆…②库…③董… Ⅲ.软件-测试 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字(2008)第 151975 号

责任编辑:任 静 张艳芬/责任校对:朱光光

责任印制:赵 博/封面设计:无极书装

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

http://www.sciencep.com

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2009 年 1 月第 一 版 开本:B5(720×1000)

2009 年 1 月第一次印刷 印张:16

印数:1—3 000 字数:315 000

定价:45.00 元

(如有印装质量问题,我社负责调换〈新蕾〉)

序　言

在开发人员和测试人员看来，软件测试是非常有趣且不同寻常的一件事。软件中的缺陷曾经导致飞机失事、空间飞行器任务失败、股票交易所暂停，乃至危及生命等各种灾难。因此，软件测试至关重要。例如，针对千年虫（Y2K）问题出现了大量的专业人员和专门工具，以避免现代社会在下个千年到来的第一天陷入混乱之中。软件调试可以用来寻找设计缺陷，但它只是软件测试的一个局部表现。软件测试活动在可信软件的开发中扮演着重要角色，它需要耗费大量的资源，包括时间、资金和人力。软件测试的困难源于软件本身的复杂性，即使一个中等复杂度的软件都很难被完全测试。软件测试是软件开发过程的一个组成部分，它耗费了软件开发过程中约40%的付出。因此，亟须当今的软件工程师深入理解软件测试，而不是仅仅盼望着他们在花费极高的实践代价后，通过经验积累来掌握软件测试。

主要特点

本书主要关注工程实践中的软件质量及保证，以满足读者的需要。与其他书籍相比，本书的主要特点包括：

（1）读者范围广泛。不像其他软件测试书籍主要以软件测试从业者为目标，本书在开始撰写时定位的目标读者就包括了学生、研究人员和工程应用人员。

（2）拓展软件测试理念。在本书中，将软件测试看做是各阶段中一个必不可少的活动，而不是仅仅作为单独的、在开发后实施的活动。

（3）通过制订目标来引导学习。书中每一章首先给出一组预期要达到的目标，以便读者在学习过程中确定正确的方向。

（4）习题。每一章之后都给出一系列习题，其中包括了客观题、简答题和启发性论述题三种类型。

（5）参考文献。每章之后针对该章中的概念给出一些重要的参考文献，以便读者获得与所学内容相关的更多、更深入的信息。

（6）相关链接。每章之后给出相应的网址列表，以便读者查找相关主题的更多资料。

本书结构

本书共分为十一章，具体组织如下：

第一章：软件测试基础

本章讲述软件开发过程中测试以及软件测试的重要性与意义，定义了软件测试的目标、特性、构成和类型，确立了软件测试在软件质量保证中的角色，阐述了软件测试与软件可靠性之间的关系，列举了软件测试中需要考虑的因素，说明了软件测试的许多特点，并给出软件测试定义的不同观点。

第二章：软件可测试性

本章把软件可测试性作为软件质量中的一个重要因素进行讲述，说明了软件可测试性度量的重要性和意义，阐述了软件可测试性度量所要考虑的各种软件特征，描述了软件可测试性因素及其作用。

第三章：静态测试

本章讲述了软件测试策略的重要性和意义，通过描述软件验证和确认过程的相关因素，定义了软件测试的方法策略。此外，本章给出了静态分析的各种原理，以及在软件测试中执行走查和审查的过程。最后说明了软件测试中技术评审的重要性和意义。

第四章：黑盒测试

本章阐述了在项目中使用黑盒测试的好处，列出了黑盒测试技术的优缺点，给出边界值分析法的目标与能力，并定义了边界条件。此外，本章讨论了健壮性测试、语法测试和有穷状态测试。

第五章：白盒测试

本章介绍了白盒测试技术，阐述了基本路径测试的重要性和主要方法。进一步讨论了控制流图的角色和目标，说明了软件测试中使用图矩阵的好处，并定义了控制结构测试。此外，还介绍了测试面向对象软件时使用的变异测试方法。

第六章：软件测试策略

本章主要讲述了常见的软件测试策略，阐述了测试策略的重要性和意义。进一步描述了软件测试策略的一般特征，以及单元测试、集成测试、确认测试和系统测试等不同策略的用途。此外，给出了测试完成的准则，最后说明了软件构件测试的重要性和意义。

第七章：软件测试计划

本章讲述了测试软件之前建立软件测试计划、测试规范中文档化的重要性。此外，详细讨论了测试成功的标准，以及如何确定哪些方面需要测试而哪些不需要测试。最后列出了测试需要交付的相关文档和产品。

第八章：面向对象测试

本章介绍了面向对象技术和面向对象软件测试方法的重要性和意义，描述了对软件测试和软件质量起到重要作用的软件特征。讨论了面向对象软件测试的原理，给出面向对象方法和传统方法的主要不同之处。

第九章：软件容错

本章讲述了容错的概念、重要性和意义，并对软件失效进行了定义和描述。介绍了错误检测中各种检测方法的相关因素，描述了不同的软件容错方法。此外，对软件错误检测的不同看法与观点也进行了介绍。

第十章：变异测试

本章主要讲述在软件开发生命周期的早期，对面向对象软件进行变异测试的重要性和意义。另外，对设计阶段的变异进行了定义，给出测试数据集的充分性评价方法，并阐述了在评价软件测试充分性时进行变异操作的重要性。

第十一章：类测试的复杂性

本章主要讲述面向对象的度量，并指出在早期使用SDLC进行度量是对软件开发进行成功管理的重要因素，也是降低类测试复杂性和成本的主要因素。本章还给出了一组度量准则，用来预测面向对象软件类测试的复杂性，并进一步介绍了利用回归线来描述软件特征与复杂度之间相关性的方法。

K. Mustafa and R. A. Khan

目　录

第一章　软件测试基础

It's better to know some of the questions than all of the answers.
James Thurber

本章目标

通过本章的学习，读者应达到以下目标：

- 了解并能描述什么是软件测试。
- 理解长期以来关于软件测试的不同定义。
- 了解软件测试的用途、重要性和意义。
- 能够列出软件测试的目标和属性。
- 知道不同类型软件测试之间的关系，并能进行比较。
- 知道软件测试在软件质量改进中所处的角色。
- 能够区分软件测试与其他类似的活动。
- 认识到软件测试所面临的各种挑战。
- 了解并能给出软件测试的主要特征。
- 了解软件测试的不同观点。
- 理解并能描述软件测试的基本原理。
- 掌握与测试相关的各种术语定义。
- 了解软件测试的有关原则。
- 了解软件测试的各种传言和误解，及其相关事实。
- 知道有效测试的重要性和意义。
- 了解软件测试成本估计方法。
- 知道调试与测试在软件生命周期中的重要性和作用。
- 知道对软件缺陷（bugs）的不同看法。
- 能够把软件测试的各种基本概念关联起来进行理解。

1.1　简　　介

大型软件产品的开发包括多种活动，这些活动需要进行合理地安排以满足期望的需求。由于软件设计与开发人员可能出现错误，以及软件本身具有抽象性和

复杂性，软件开发必须伴随着一系列的质量保证活动。对于开发者，即使花费整个项目时间的40%进行测试也是很常见的。测试通常是一个极其昂贵且难以控制的过程，可能需要超过预期计划的大量时间和代价，最终还无法对测试过程的质量进行充分的评价。测试过程包括计划、准备、执行，以及对软件实际状况与预期之间差异进行分析等活动。软件测试是对软件产品或程序进行验证和确认的过程，以确定软件是否满足对设计和开发起导向作用的业务或技术需求，以及软件是否像期望的那样工作。测试计划和准备活动强调测试不只是一个在被测软件实现后才开始的工作。测试计划通过审查需求和设计文档，根据决定软件使用性和功能性的客户观点，确定哪些可能是软件的重要缺陷。一个测试过程需要在实施任何具体措施前进行准确的计划和准备。测试并不能表明软件不存在缺陷，但它可以明确表现出某些缺陷的存在。

到目前为止，已出现了多种软件测试的定义，这些定义都有一定的道理，但表述方式不尽相同。其中一部分定义关注于测试中要做哪些事情，而另一部分定义则关注于质量评价、客户满意度、预期结果等更加一般的目标。

1973年，Hetzel在其著作 *Program Test Methods* 中指出：测试是建立关于一个程序按照预期目标运行的可信度的过程。该定义关注于客户的满意度。

1979年，Myers在其经典著作 *The Art of Software Testing* 中认为：测试是为了发现错误而执行一个程序或系统的过程。在当时，Myers的定义也许是最能反映当时人们想法的。该定义所关注的事实是测试在软件生命周期的后期实施，它的主要目的是发现错误。

1983年，Hetzel在 *The Complete Guide to Software Testing* 中提出：测试包括评价一个程序或系统某个属性的任何活动，测试是对软件质量的度量。

软件测试是验证一个系统是否满足规定需求或识别实际结果与预期之间差异的过程。在计算机领域中，测试定义为：使用一组受控的条件和激励，以发现错误为目标的验证方法。它是验证功能和性能需求最常用的方法。测试结果需要文档化以表明需求已被满足并可以进行复现。结果数据可以被所有相关人员审查以证实其效力。

美国国家标准学会（ANSI）在其官方标准中使用了两个不同的定义，它们都与满意度及确认相关：

① 在特定条件下操作一个系统或构件，观察或记录其结果，并对系统或构件的某些方面进行评价的过程（IEEE/ANSI，1990［Std 610.12-1990］）。

② 分析一个软件项以检测目前与所要求的条件之间的差异（即缺陷），并对软件项的特征进行评价的过程（IEEE/ANSI，1983［Std 829-1983］）。

现今，软件规模已变得日趋庞大和复杂，在看待一个软件产品时，应该考虑其整体而不是仅限于程序。软件由程序、数据和文档组成，如图1.1所示。

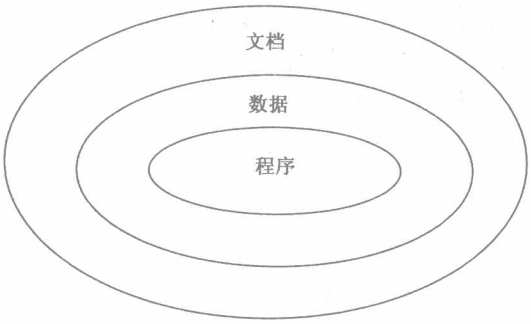


图 1.1　软件的组成

由此可见，以一种局限的方式进行软件测试，即仅对程序进行测试是远远不够的，应更新测试理念，把测试范围拓宽到所有的软件组成。软件测试是系统开发必不可少的一部分，其具有以下的特征：

① 软件测试可以从需求开始，而不仅仅是代码。

② 软件测试既是静态活动也是动态活动。

③ 软件测试用来预防失效的出现。

④ 软件测试有助于在软件生命周期中尽早发现问题，以降低修复缺陷所需的成本。

⑤ 软件测试过程应创建可重用的测试件。

软件测试是一个团队活动，而不是单独一个人的工作。团队的规模可以根据被测应用软件的复杂度进行调整。在测试过程中可以适当减少编码程序员的参与程度，因为他们可能难以用一种不带偏见的眼光对待其劳动成果。测试人员应具备谨慎、好奇、批判（但不是审判）的态度和善于交流的能力。

1.2　软件测试概述

软件测试代价一般都很高，因此在软件生命周期内经常得不到应有的重视。软件开发过程需要特别关注如何避免错误，如何检测和更正发生的软件故障，以及在开发之后预测软件的可靠性。人们相信软件工业面临引发某种灾难的风险，而这种灾难是由软件导致的。许多计算机系统用于关键应用领域，例如航天器和国防系统。当生命和财产依赖于软件时，软件质量及其验证就得到了极大地关注。随着对软件质量和客户满意度要求的不断提高，需要对测试的目标和管理重新进行思考。软件测试在发现缺陷中显然扮演着重要的角色，但在可靠性评价中的重要性还不是很明显。测试和评价方法以及工具本身并不足以保证测试的有效

性和软件的质量。由于软件及其应用的复杂性不断提高，软件测试和评价变得更加困难，产生的效果也低于人们的预期。在过去 20 年中，软件测试技术和方法学、测试用例生成、测试辅助工具等有了长足地发展。

软件测试的基础包括：测试过程、测试用例和测试计划；技术、方法学、工具和标准；人和组织机构。如图 1.2 所示。

图 1.2 软件测试基础

1.2.1 相关术语

软件测试至今已经建立起一套专业术语，软件测试过程和相关文献中常用的技术术语简要描述如下：

审计（audit）：对一个或一组工作产品进行独立检查，以评价它们是否遵从有关规格说明、标准、合同条约或其他准则。

边界值（boundary value）：一个系统或构件的输入、输出或内部变量所规定的最大值或最小值。

边界值分析（boundary value analysis）：一种测试数据选择技术，使测试数据取值位于输入（或输出）的类、数据结构、过程参数等元素的边界。其选择通常包括最大值、最小值、平凡值等。

分支覆盖（branch coverage）：一种测试覆盖准则，要求每个决策点的各个可能分支都至少执行一次。

缺陷（bug）：程序中存在的错误或故障，将导致程序以一种非预期的方式运行。

崩溃（crash）：计算机系统或构件突然地、完全地失效。

危险程度（criticality）：需求、模块、错误、故障、失效或其他方面对一个系统的开发或操作产生影响的程度。

圈复杂度（cyclomatic complexity）：一个程序中独立路径的数目，等于分支语句的数目加 1。

错误（error）：一个值或条件经过计算、观察或度量所得到的结果与真实的、规定的或理论上的正确结果之间存在的差异。

错误猜测（error guessing）：一种测试数据选择技术，其准则是选择那些看起来更可能引发错误的值。

错误植入（error seeding）：故意向计算机程序中添加已知故障的过程，其目的是判断故障被检测到和被删除的比率，然后估计程序中依然残留的故障数量。

异常（exception）：一个导致程序正常执行被中止的事件。其类型包括地址异常、数据异常、操作异常、上限溢出异常、保护异常和下限溢出异常等。

穷尽测试（exhaustive testing）：用程序变量取值的所有可能组合来运行该程序。该测试方式只适用于小规模、简单的程序。

失效（failure）：一个系统或构件在规定的性能指标下，失去执行所要求的功能的能力。

故障（fault）：计算机程序中不正确的步骤、过程或数据定义，能够导致程序以一种非预期的方式运行。

预言（oracle）：用于预测测试结果的方法。

风险（risk）：对不期望出现的结果将会发生的概率和严重程度的度量。

风险评估（risk assessment）：对风险及其相关影响进行全面的评价。

测试（test）：在特定条件下运行一个系统或构件并进行观测和记录，以评价系统或构件的某些方面。

可测试性（testability）：为一个系统或构件建立测试准则以及根据测试结果判断这些准则是否被满足的难易程度。

测试用例（test case）：用来为一个测试项指定输入、预期结果和相关的运行条件。

测试用例生成器（test case generator）：以源代码、测试准则、规格说明或数据结构定义作为输入，然后生成测试输入数据并确定预期结果的软件工具。

测试设计（test design）：针对一个软件特征或软件特征的组合，规定具体的测试方法并给出相关检验方式的过程。

测试文档（test documentation）：描述测试一个系统或构件的计划或结果，其类型包括测试用例规格说明、测试事件报告、测试日志、测试计划、测试步骤和测试报告等。

测试驱动（test driver）：用来调用一个被测模块的软件模块，它通常提供测试输入，控制和检测运行过程并报告测试结果。

测试事件报告（test incident report）：对测试期间发生并需要进一步调查的任何事件进行报告的文档。

测试项（test item）：被测对象的软件构件或模块。