免费提供电子教案

高等院校规划教材
软件工程系列

软件测试技术

路晓丽　葛 玮　龚晓庆　等编著

机械工业出版社
CHINA MACHINE PRESS

软件测试技术

高等院校规划教材·软件工程系列

软件测试技术

路晓丽　葛 玮　龚晓庆　等编著

机 械 工 业 出 版 社

本书全面、系统地论述了软件测试的理论和应用技术。全书共3部分，其中，第1部分(第1～9章)介绍了软件测试的基本理论和测试用例的设计方法，包括软件测试的概念、白盒测试、黑盒测试、自动化测试、性能测试、兼容性测试、安全性测试、特定环境及应用测试等测试基础知识和应用技术。第2部分(第10～14章)介绍了面向对象软件的测试，包括面向对象软件分析和设计模型的测试、类测试、交互测试和系统测试等。第3部分(第15～17章)介绍了测试管理的基本知识，包括测试文档和测试计划、测试项目的管理、测试小组的管理等。

本书可以作为大学本科软件测试课程的教材，也可以作为软件测试人员、软件项目经理和需要了解软件测试的各级管理人员的参考书。

图书在版编目（CIP）数据

软件测试技术/路晓丽等编著. —北京：机械工业出版社，2007.8
（高等院校规划教材·软件工程系列）
ISBN 978-7-111-22155-5

Ⅰ.软… Ⅱ.路… Ⅲ.软件－测试－高等学校－教材 Ⅳ.TP311.5

中国版本图书馆CIP数据核字（2007）第124457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：赵 慧
责任印制：洪汉军

北京振兴源印务有限公司印刷厂印刷

2007年9月第1版·第1次印刷
184mm×260mm·18.75印张·460千字
0001—5000册
标准书号：ISBN 978-7-111-22155-5
定价：27.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：(010)68326294
购书热线电话：(010)88379639 88379641 88379643
编辑热线电话：(010)88379739
封面无防伪标均为盗版

出 版 说 明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。另外，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

随着软件产业的更新换代和软件企业规模的扩大，软件测试的重要性日益凸显，软件测试行业在国内渐渐兴起。软件测试（Software Testing）作为软件工程学科的一个重要分支，随着软件的发展而发展，已经成为软件质量保证的关键技术之一，也是软件开发过程中的一个重要环节。软件测试工作做得怎样，直接决定着软件产品的质量。

目前，我国 IT 产业对软件测试人才的需求日益增涨。根据国家信息产业部提供的数据显示，目前软件测试人才的缺口已经达到 30 万至 40 万，软件测试人才正在成为我国 IT 行业最紧缺的人才之一，软件测试能力不足已成为制约我国软件产业发展的重要因素。

本书全面、系统地论述了软件测试的理论知识和应用技术，介绍了测试理论、测试方法、测试技术、测试工具和测试管理等知识，涵盖了业界出现的大部分测试领域内的知识，是一本比较全面的介绍测试知识的书籍。同时，本书在编写时注意理论结合实际，书中许多涉及理论的内容都使用例子进行阐述和说明，是一本非常实用的软件测试教材。

全书共 3 部分，其中，第一部分（第 1～9 章）介绍了软件测试的基本理论和测试用例的设计方法，包括软件测试的概念、白盒测试、黑盒测试、自动化测试、性能测试、兼容性测试、安全性测试、特定环境及应用测试等测试基础知识和应用技术。第二部分（第 10～14 章）介绍了面向对象软件的测试，包括面向对象软件分析和设计模型的测试、类测试、交互测试和系统测试等。第三部分（第 15～17 章）介绍了测试管理的基本知识，包括测试文档和测试计划、测试项目的管理、测试小组的管理等。每章后附有习题，供读者练习。

本书总结了作者多年从事软件测试课程教学的经验和在软件公司从事软件测试工作的工作经验，同时参阅了大量国内外相关文献资料，不断地进行总结和充实，完成了教材书稿的编写工作。本书可以作为大学本科软件测试课程的教材，也可以作为软件测试人员、软件项目经理和需要了解软件测试的各级管理人员的工作参考书。对于那些希望增强软件测试方面知识的程序员及软件开发团队的其他人员，本书也具有很好的参考价值。

软件测试作为一门软件工程方面的学科，从软件工程中演化出来，并且还在不断发展之中。掌握一门高级语言，并且了解一定的软件开发过程知识对于本书的理解是十分有益和必要的。本书作为高等院校计算机软件测试技术的教材，体现了计算机软件技术课程改革的方向之一。本课程建议授课学时为 54 小时，实验学时 18 小时（折合 36 机时），并要求先学软件工程和一门高级语言，例如：Visual C++ 或者 Java 语言。

本书主要由路晓丽、葛玮、龚晓庆编写，参加编写工作的还有刘海、李红红、王莉、周庆泉、郭强等。本书的顺利出版，要感谢机械工业出版社，以及西北大学的领导和老师给予的大力支持和帮助。

对于书中存在的不妥之处，敬请有关专家和读者提出宝贵意见。

编者

目　录

第2部分 面向对象的软件测试

第 3 部分 软件测试管理

第 1 部分　软件测试基础

第 1 章　软件测试概述

本章介绍软件危机的内在原因及软件工程的解决方法、软件可靠性和质量特性、软件测试技术的基础知识、软件测试职业和职位等基础理论知识。这些知识是学习本书后续内容的必要准备。

1.1　软件危机

1.1.1　什么是软件危机

随着计算机技术的飞速发展，计算机已广泛应用于许多领域。为了适应在广泛应用情况下出现的各种各样的复杂问题，需要严格保证软件系统的质量。但是，软件系统的开发需要投入大量的人力、物力和财力，相对于硬件系统投资而言，软件投资所占比例越来越大。同时，开发软件本质上是一个“思考”的工程，开发人员有各自的编程习惯和思维方式，可凭个人的爱好进行工作，没有统一的标准可以遵循，因此系统的质量难以得到保证。大约在 20 世纪 60 年代，面对愈来愈复杂的大型软件系统开发，出现了“软件危机”，主要表现为以下几个方面：

- 软件项目经常无法按期完成，超出经费预算，软件质量难以控制。
- 开发过程管理不规范，约定不严密，文档书写不完整，使得软件维护费用高，有些系统甚至无法进行修改。
- 缺乏严密有效的质量检测手段，交付给用户的软件质量差，在运行中出现许多问题，甚至带来严重的后果。
- 系统更新换代难度大。

这些问题出现在很多具体的软件开发项目上，最为突出的实例是美国 IBM 公司在 1963 年至 1966 年开发的 IBM 360 机操作系统。这一项目在开发期间每年花费 5000 万美元，总共投入的工作量为 5000 人/年，参加工作最多时有 1000 人，总共写出了 100 万行源代码。这么大的开销，却拿不到开发成果，这是一次失败的记录。项目负责人 Brooks 事后总结了他在开发过程中的沉痛教训，写成了《人月神话》一书。这个反映软件危机的典型事例已成为软件技术发展过程中一个历史性标志。

1.1.2　软件危机的内在原因

在软件系统的开发过程中，软件缺陷的积累与放大效应是导致软件危机的最主要原因，可谓失之毫厘，差之千里。在此情况下，反复无常的修改导致软件开发效率严重低下，带有缺陷

的开发成果导致软件质量大幅度下降，不断投入的人员和其他资源导致软件开发成本急剧增加。因而，迫切需要有规范化的工程来制约软件开发的无序性，即像处理“工程”一样来处理软件研制的全过程。1968 年，在北大西洋公约组织的学术会议上，第一次提出了“软件工程”一词，倡导按照工程化的原则和方法组织软件开发工作。

1.1.3 软件工程和软件危机的解决

软件工程主要是通过提供规范化的分析设计方法及相应的工具软件，来避免或减少软件错误的发生，为最终根除软件危机提供强有力的技术保障。在软件工程中，为了确保质量，软件的含义已不再仅仅是指程序了，而明确为“软件是程序以及开发、使用和维护程序所需的所有文档”；“研制软件”也不仅仅是“编程序”了，而明确为“研制软件过程中所涉及到的所有活动，包括分析、设计、编码、测试和维护等”。

当用工程化的方式有效地管理软件开发的全过程时，程序的编写只是整个工作的一部分，在它的前后还有更重要的工作。一般来说，任何计算机软件都有它的生命周期，一般可分为 5 个阶段：需求分析(Requirement Analysis)、设计(Design)、程序编写(Coding)、测试(Testing)、运行和维护(Run and Maintenance)。每个阶段都有明确的任务，并生成一定规格的文档交送给下一个阶段。

从表 1-1 中可以看出，开发期中各阶段的工作量是不一样的，软件测试占有非常突出的地位，工作量及开销都要占一半左右，软件测试成为保证软件质量的重要手段，越来越引起人们的重视。

表 1-1 软件工程各个阶段的基本情况

阶段		基本任务	工作成果	占开发期的工作量	参加者
开发期	需求分析	理解和表达用户的要求、对开发的软件进行详细的定义	系统需求说明书	20%	用户、系统分析员、高级程序员
	设计	概要设计和详细设计，建立系统的结构，明确系统的实现方式	系统设计说明书、数据说明	15%	系统分析员、高级程序员
	程序编写	写程序	程序	20%	高级程序员、初级程序员
	测试	发现错误和排除错误	可运行的系统	45% (模块测试 25% 其他测试 20%)	测试工程师、测试员
运行期	运行和维护	维护	改进的系统		用户、程序员

1.2 软件质量与可靠性

1.2.1 软件质量因素和质量特性

计算机软件是非常复杂的产品，软件质量主要表现为三个方面：第一，软件需求是衡量软件质量的基础，符合用户需求的软件才是高质量的软件；其次，软件开发必须遵循规定的标准；第三，软件应该满足那些隐含的要求，否则软件产品的质量仍然是有问题的。

影响软件质量的因素分为两类:一类是可直接度量的因素,例如,单位时间内每千行源代码所发现的错误个数;另一类是只能间接度量的因素,例如,可复用性、可维护性等。不论哪一类,都必须能够度量,都可以以具体数据表达软件质量的不同方面。软件的质量因素可以从不同的方面反映软件的质量要求,软件的质量因素分为三类,分别为:软件的运行特性、软件的修正特性和软件的转移特性。

(1) 软件的运行特性

- 正确性:软件能满足其规格说明及完成客户提出任务要求的程度。
- 可靠性:该软件能够总是精确地工作吗?
- 有效性:该软件是否能在计算机上有效地运行,内存、外设容量是否够用?
- 完整性:控制未经允许人员使用软件或数据的能力。
- 可用性:此软件容易掌握吗?

(2) 软件的修正特性

- 可维护性:找到软件错误发生的位置并加以修正所花费工作量的大小。
- 灵活性:可以修改,容易修改吗?
- 可测试性:可以测试,容易测试吗?

(3) 软件的转移特性

- 可移植性:可以在别的机器上使用这个软件吗?
- 可复用性:它能重复使用吗?
- 共运行性:可以把这个软件与别的软件对接吗?要能做到这样得花多少工作量?

以上这些质量因素往往难于,甚至不可能直接度量,因此必须进一步将质量因素分解成一些独立的、容易度量的质量特性(Quality Characteristics)。质量特性为质量提供了更完全、更具体的定义,有助于说明各质量因素相互之间的关系,准确地决定质量因素的范围,方便质量检查和质量的定量观测。

1.2.2 软件可靠性

IEEE 对软件可靠性的定义:系统在特定环境下,在给定的时间内无故障运行的概率。

软件可靠性是对软件在设计、开发以及所预定的环境下具有能力的置信度的一个度量,是衡量软件质量的主要参数之一。软件可靠性主要关注软件是否可以稳定地工作,可靠性差意味着软件在执行时总是频繁地、重复地失败,软件不能稳定工作。软件可靠性可以用工作历史数据和开发数据来测量、标示和估算出来。

考虑软件可靠性时,需要注意四个方面:①研究的对象是软件,而不是一般系统或产品,软件特有的一些性质应考虑在内;②注意规定的功能或失败的含义,因为任何软件开发出来都是要求它完成某些特定的功能,在软件开发完成、投入运行以后,如果出现了异常现象,就是发生了故障(Failure),一旦程序出现了故障,其中必定隐藏着相应的缺陷(Fault);③注意软件的工作环境,在不同的环境条件下和不同的应用领域中,对于故障的定义常常有很大的区别;④注意规定的时间,例如次数、周期、距离等相当于时间的量,以及连续使用、间歇使用、放置、长时间、短时间、瞬间等各种时间概念。软件可靠性的简单度量是"平均失败间隔时间"。

对软件系统的可靠性进行估测,大致有以下几个步骤:

1) 根据软件开发过程及以前开发的类似软件的可靠性,给出关于它们的"先验"可靠性。

2）收集数据。在尽可能符合软件使用时的条件和环境下，对软件进行测试；尽可能多地收集并记录进行预测所需的精确数据。软件故障数据的缺乏是目前软件可靠性研究中颇具制约性的因素。

3）选择符合该软件特性及测试情况的统计模型。

4）利用收集到的数据估计模型中的参数。

5）对软件可靠性及程序的行为进行预测。

1.2.3 软件错误

软件错误是指软件没有实现其最终用户合理预期的功能要求。软件错误除了包括程序编写中的错误，例如，数组越界、变址或移位差 1、用错特征位、补码运算错、使用指针的问题、控制转移的问题以及间接寻址出错的问题等，还包括了很多类型的错误。从软件错误的性质看，可以把软件错误分为以下几种类型，如表 1-2 所示。

表 1-2 软件错误分类表

错　误	含　义
软件需求错误	软件需求制定得不合理或不正确；需求不完全；其中含有逻辑错误；需求分析的文档有误等
功能和性能错误	功能或性能规定得有错误，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；对异常处理有误等
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
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，例如动态数据与静态数据混淆、参数与控制数据混淆等
软件实现和编码错误	编码或按键错；违背编码风格要求或是编码标准，包括语法错、数据名错、局部量与全局量混淆，或是程序逻辑有误等
软件集成错误	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
软件系统结构错误	操作系统调用错或使用错、恢复错误、诊断错误、分割覆盖错误引起环境的错误等
测试定义与测试执行错误	测试的错误往往不被人们重视，它可能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

表 1-2 中的软件错误中最值得重视的是软件结构错误、数据错误、功能和性能错误，因为这三种错误最为普遍。

1.3 软件测试的基本知识

1.3.1 软件测试的背景和意义

软件测试在软件生存期中占有非常突出的位置，是保证软件质量的重要手段。软件项目的实践一再说明，为了确保软件产品能够符合用户的需要，必须着眼于整个软件生存期，在各个阶段进行验证、确认和测试活动，使软件不致在开发完成后，才发现和用户的需求有较大的差距。

软件在很多领域广泛使用，然而软件是人编的，难免存在各种各样的缺陷。下面给出几个历史上的著名案例。

案例 1：迪斯尼的狮子王多媒体光盘，1994-1995

1994 年秋天，迪斯尼公司发布了第一个面向儿童的多媒体光盘游戏 Lion King Animated Storybook(狮子王动画故事书)。公司进行了大力宣传，销售额非常可观。但是，同年 12 月 26 日，迪斯尼公司的客户服务部就淹没在愤怒的家长和孩子的电话狂潮中。后来证实，迪斯尼公司并没有对市场上投入使用的各种 PC 机型进行正确的测试，软件在少数系统中工作正常，但在大众使用的常见系统中却不能正常工作。

案例 2：美国航天局火星极地登陆飞船，1999

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失踪。错误修正委员会观测到故障，并认定出现误动作的原因极可能是某一个数据位被意外修改。大家一致声讨问题为什么没有在内部测试时解决。后来调查发现，登陆飞船经过了多个小组测试，其中一个小组测试飞船的落地位置，另一个小组测试此后的着陆过程。前一个小组没有注意着地数据位是否置位，而后一个小组在开始测试之前重置计算机、清除数据位，双方独立工作都很好，但却从未协调在一起测试。

案例 3：爱国者导弹防御系统，1991

美国爱国者导弹防御系统首次应用在海湾战争中对抗伊拉克飞毛腿导弹的防御战争中，几次失利，其中一枚在多哈地区击毙 28 名美军士兵。分析专家发现问题在于一个软件缺陷，一个很小的时钟错误积累起来就可能拖延 14 小时，造成跟踪系统失去准确度。

案例 4：千年虫，1974

20 世纪 70 年代的一位程序员，为了节省空间，开发工资系统时将 4 位日期缩减为两位，致使类似系统在 2000 年到来之前，更换或升级系统以解决 2000 年错误的费用超过了数亿美元。

案例 5：Intel 奔腾浮点除法，1994

在计算器中输入以下算式：(4195835/3145727)×3145727-4195835，如果结果为零，则计算器没有问题，若不为零，则使用的是老式 Intel 芯片。

以上只是一部分软件失败时发生的历史事件，后果也许是不方便使用，也可能是灾难性的。在这些事件中，显然软件未按照预期目标运转，软件出现了软件缺陷。随着时间的推移，软件缺陷修复的费用会数十倍地增长，例如，若编写需求说明书时就发现了软件缺陷，费用可能只要几角钱；若在测试时才发现软件缺陷，费用可能要几元钱；若缺陷是客户发现的，费用可能达到几百元。比如“迪斯尼狮子王”案例，假如在编写需求说明书时，项目成员已经研究过什么 PC 机流行，并且明确指出软件需要在该配置上设计和测试，付出的代价非常小；如果没有这样做，就需要软件测试员去搜集流行 PC 样机并在其上验证，可能会发现软件缺陷，这时付出的代价要高得多。

因此，随着当今软件规模和复杂性的日益增加，进行专业化高效软件测试的要求越来越迫切，挑战性极强。软件测试员的目标就是找出缺陷，且尽可能早一些，并确保其得以修复，从而保证软件的质量。

那么，到底什么是软件缺陷呢？以下给出符合软件缺陷的五个规则：

- 软件未达到产品说明书中已经标明的功能。
- 软件出现了产品说明书中指明不会出现的错误。
- 软件未达到产品说明书中虽未指出但应当达到的目标。

- 软件功能超出了产品说明书中指明的范围。
- 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良。

例如,某计算器的产品说明书可能明确地声明了它能够准确无误地进行加、减、乘、除运算。假如一个测试人员,按下"+"键时什么反应也没有,根据第一条规则,这是一个软件缺陷;假如得到了错误的答案,根据第一条规则,仍然是软件缺陷。产品说明书中声明计算器永远不会崩溃或者停止反应,但假如连续敲击键盘使得计算器不能接受输入,则根据第二条规则,这是一个软件缺陷。测试计算器时,电池没电或者电量不足会导致计算不正确,根据第三条规则,这是软件缺陷;假如计算器除了加减乘除外,还可以求平方根,这一功能产品说明书中并没有说明,根据第四条规则,这是软件缺陷;测试人员使用不便的地方,例如按键小、布局不好,根据第五条规则,这些也是软件缺陷。

1.3.2 软件测试的定义

软件测试在软件开发成本中占有很大的比例,是保证软件质量的主要手段,越来越受到人们的重视。那么,什么是软件测试呢? 这一基本概念很长时间以来存在着不同的观点。

Glen Myers 认为"程序测试是为了发现错误而执行程序的过程"。这一定义明确指出"寻找错误"是测试的目的。相对于"程序测试是证明程序中不存在错误的过程",Myers 的定义是对的。把证明程序无错当作测试的目的不仅是不正确的,也是完全做不到的,而且对做好测试工作没有任何益处,甚至是十分有害的。从这方面讲,我们接受 Myers 的定义以及它所蕴涵的方法论和观点。不过,这个定义规定的范围似乎过于狭隘,使得它受到很大限制。因为如前所述,除去执行程序以外,还有许多方法去评价和检验一个软件系统。

另外,有些测试专家认为软件测试的范围应当包括得更广泛些。J.B.Goodenough 认为测试除了要考虑正确性以外,还应关心程序的效率、健壮性(Robustness)等因素,并且应该为程序调试提供更多的信息。S.T.Redwine 认为,软件测试应该包括几种测试覆盖,分别为:功能覆盖、输入域覆盖、输出域覆盖、函数交互覆盖、代码执行覆盖。关于测试的范围,A.E.Westley 将测试分为四个研究方向,即:验证技术(目前验证技术仅用于特殊用途的小程序)、静态测试(应逐步从代码的静态测试往高层开发产品的静态测试发展)、测试数据选择、测试技术的自动化。

总的来说,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实现的最终审查,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:

定义 1:软件测试是为了发现错误而执行程序的过程。

定义 2:软件测试是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的一批测试用例,并利用这些测试用例运行程序以发现错误的过程。

在 IEEE 提出的软件工程标准术语中,软件测试被定义为:"使用人工和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楚预期结果与实际结果之间的差别。"

事实上,所有发布的软件产品都会因为缺陷而导致用户的困扰和开发者时间和金钱上的额外开支。而这些导致成本风险的软件问题可以通过在软件生命周期的每一个阶段中充分规划和执行验证和确认(Verification and Validation)而大大降低。

广义的软件测试是由确认、验证、测试三个方面组成。

- 确认：是评估将要开发的软件产品是否是正确无误、可行和有价值的。这里包含了对用户需求满足程度的评价，意味着确保一个待开发软件是正确无误的，是对软件开发构想的检测。
- 验证：是检测软件开发的每个阶段、每个步骤的结果是否正确无误，是否与软件开发各阶段的要求或期望的结果相一致。验证意味着确保软件正确无误地实现软件的需求。
- 测试：与狭隘的测试概念统一。通常是经过单元测试、集成测试、确认测试和系统测试四个环节。

在整个软件生存期，确认、验证、测试分别有其侧重的阶段。确认主要体现在计划阶段、需求分析阶段，也会出现在测试阶段；验证主要体现在设计阶段和编码阶段；测试主要体现在编码阶段。事实上，确认、验证、测试是相辅相成的，确认无疑会产生验证和测试的标准，而验证和测试通常又会帮助完成一些确认，特别是在系统测试阶段。因此，软件测试贯穿于软件定义和开发的整个过程。软件开发过程中所产生的需求规格说明、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。

1.3.3 软件测试的分类

软件测试按照不同的划分方法，有不同的分类。按照程序是否执行，可以分为静态测试和动态测试；按照测试用例的设计方法，可以分为白盒测试和黑盒测试；按照开发阶段划分，可分为单元测试、集成测试、确认测试、系统测试和验收测试；按照测试实施组织划分，可分为开发方测试、用户测试(β测试)和第三方测试；按照是否使用工具软件，可以分为手工测试和自动测试。

1. 静态测试和动态测试

(1) 静态测试

原则上讲，可以把软件测试分为两大类，即静态测试和动态测试。静态测试的主要特征是在用计算机测试源程序时，计算机并不真正运行被测试的程序。这说明静态方法一方面要利用计算机作为对被测程序进行特性分析的工具，它与人工测试有着根本的区别；另一方面它并不真正运行被测程序，只进行特性分析，这是和动态方法不同的方面。因此，静态测试常称为“分析”，静态分析是对被测程序进行特性分析的一些方法的总称。

值得注意的是，静态分析并不等同于编译系统，编译系统虽也能发现某些程序错误，但这些错误远非软件中存在的大部分错误，静态分析的查询和分析功能是编译程序所不能代替的。目前，已经开发出一些静态分析系统作为软件测试的工具，静态分析已被当作一种自动化的代码校验方法。不同的方法有各自的目标和步骤，侧重点不同。

静态测试阶段的任务主要表现为以下方面：检查算法的逻辑正确性；检查模块接口的正确性；检查输入参数是否有合法性检查；检查调用其他模块的接口是否正确；检查是否设置了适当的出错处理；检查表达式、语句是否正确，是否含有二义性；检查常量或全局变量使用是否正确；检查标识符的使用是否规范、一致；检查程序风格的一致性、规范性；检查代码是否可以优化，算法效率是否最高；检查代码注释是否完整，是否正确反映了代码的功能。

静态测试包括代码检查、静态结构分析、代码质量度量等。它可以由人工进行，也可以借助软件工具自动进行。

1) 代码检查包括代码走查、桌面检查、代码审查等，主要检查代码和设计的一致性，代码对标准的遵循、可读性，代码的逻辑表达的正确性，代码结构的合理性等方面。代码检查的具