高等学校计算机规划教材

国家示范性软件学院系列教材

软件测试实用教程
——方法与实践（第2版）

◆ 武剑洁　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http: //www.phei.com.cn

高等学校计算
国家示范性软

软件测试实用教程

——方法与实践(第2版)

武剑洁　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是华中科技大学精品课程建设成果。本书围绕软件测试的核心概念，介绍了软件测试的基本方法和过程，并通过丰富的案例予以实践。全书共三部分。第一部分软件测试概述，对软件测试的核心概念与思想(软件缺陷、测试用例、自动化测试)展开初步的讨论和测试实践。第二部分软件测试技术，详细讨论了传统的黑盒测试方法和白盒测试方法，针对每种测试方法均按照基本原理、测试用例设计和捉虫实践的顺序依次展开阐述；对应黑盒测试和白盒测试给出了综合案例实践。第三部分软件测试应用，从测试实施的角度，分为单元测试、集成测试和系统测试三个阶段进行讨论；最后提供了综合应用案例实践，从自动化测试的角度，结合单元测试工具、功能测试工具和性能测试工具，讨论自动化测试的设计与实施。

全书结构遵循学生的认知规律，循序渐进，由浅入深，并注重理论联系实际，使理论知识在实践运用中具有可操作性。本书配有电子课件、教学大纲、案例源代码或安装包、典型文档模板等教学资源。

本书可作为高等学校计算机、软件工程专业“软件测试”课程的教材，也可作为软件测试人员的技术参考书。

图书在版编目(CIP)数据

软件测试实用教程：方法与实践 / 武剑洁编著. —2 版. — 北京：电子工业出版社，2012.11
高等学校计算机规划教材
ISBN 978-7-121-18678-3

Ⅰ. ①软… Ⅱ. ①武… 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(2012)第 239960 号

策划编辑：史鹏举
责任编辑：史鹏举
印　　刷：涿州市京南印刷厂
装　　订：涿州市京南印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1092　1/16　印张：19.5　字数：563 千字
印　　次：2012 年 11 月第 1 次印刷
定　　价：35.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

软件已渗透到每个人的日常生活，软件测试作为软件质量保证的关键内容之一，在软件质量控制上的地位不可替代。

当前，软件测试行业在国外已发展得相当成熟，国内软件在软件测试方面与国际水准相比仍存在着较大的差距，例如，重开发轻测试，不重视测试流程管理；管理随意，缺乏规范的软件测试管理体系；缺少自动化测试工具的支持(据软件测试网 2011 年对软件测试从业人员的调查表明，国内软件测试从业人员从事的测试工作类型主要集中在手动功能测试，高达 89%；而功能自动化测试和性能测试分别仅占 31%和 35%)。

软件测试工程师是目前软件行业中唯一存在缺口的职业，企业迫切需要各类院校提供满足企业需求的软件测试专门人才。国内软件测试从 2000 年开始发展至今已有 10 多年，但相关人才培养并未及时跟上市场发展的脚步。2002 年之前，国内高校鲜有软件测试课程。目前不少高院纷纷开设软件测试课程，甚至开设软件测试专业。然而，课程建设情况良莠不齐，软件测试专业仍然非常匮乏，软件测试课程的学时数仍然偏少，甚至有很多都没有单独开课，仅仅作为“软件工程”课程中的某一章节来介绍。高校每年培养和输送的专业测试人才仍远远无法满足企业的实际需求。

为了缩短高校毕业生水平与企业要求的差距，提高在校生对软件工程乃至软件测试过程的实践能力十分必要。为此，我们将科研实践和教学中积累的经验与体会按照概述、技术、应用三部分进行整理和提炼，与读者分享。

本书围绕软件测试的核心概念，讨论了软件测试的基本方法和过程，并通过丰富的案例予以实践。全书共三部分。第一部分软件测试概述，依托一个流行的第二日问题作为案例，对软件测试的核心概念与思想(软件缺陷、测试用例、自动化测试)展开初步的讨论和测试实践，使读者快速切入。第二部分软件测试技术，详细讨论了传统的黑盒测试方法和白盒测试方法，针对每种测试方法均按照基本原理、测试用例设计和捉虫实践的顺序依次展开阐述，讨论方法时选用了函数级别的小型实例，便于读者理解和掌握基本的测试技能；对应黑盒测试和白盒测试给出了综合案例实践，有助于读者根据所学的测试用例设计方法加以灵活运用。第三部分软件测试应用，从测试实施的角度，分为单元测试、集成测试和系统测试三个阶段进行讨论，使读者了解软件测试过程的展开，并学会撰写规范的测试用例报告和缺陷报告；最后提供了综合应用案例实践，从自动化测试的角度，结合单元测试工具、功能测试工具和性能测试工具，讨论自动化测试的设计与实施。

全书结构遵循学生的认知规律，循序渐进，由浅入深，并注重理论联系实际，使理论知识在实践运用中具有可操作性。本书主要特点如下：

(1) 紧跟技术发展前沿。本书除对基础知识进行分析讨论之外，还简要介绍了软件测试领域的研究热点和部分新技术及成果，有助于开拓学生视野，并给学生提供进一步研究的方向和思路。

(2) 突出案例分析和对比。案例的作用不应局限于对方法的简单验证。为此，本书在设计案例时，做到“三同三不”，即：

① 同一个案例，基于不同的需求，形成不同规模的程序，对应不同的测试阶段，便于学生分析各阶段的测试难度和工作量。例如，第二日问题具有单个函数版本、集成版本、系统版本等不同规模的多个程序版本。

② 同一个案例，使用不同测试方法进行测试，以助于学生进行测试设计。例如，第二日问题、保险金问题案例均使用了多种手工和自动化测试方法来展开测试尝试。

③ 同种测试方法，应用于不同案例，帮助学生理解测试方法的应用场合，从而能灵活运用各种方法。例如，针对边界值测试、独立路径测试等方法，应用于第二日、佣金、自动柜员机等多个案例进行测试。

(3) 强调自动化测试的设计。自动化测试必不可少，但自动化测试不等于看看用户手册、学会操作软件而已。因此，本书并未长篇大论地介绍测试工具的使用过程，而是依托案例，始终强调测试设计和自动化测试设计，即测试用例的设计应紧贴用户需求，自动化测试的设计应紧贴测试用例的需要，通过不同规模和特点的软件系统案例来展示如何根据被测系统特点，在不同阶段灵活使用不同的自动化测试工具。

读者可参考如下建议学习本书：

(1) 阅读第 1 章，通过亲自动手实践，初步理解什么是软件测试，如何做软件测试，如何引入自动化测试来提高测试效率。

(2) 阅读第 2 章，了解软件测试的发展历程、职业前景，扩展自己的视野，找到自己的兴趣点。

(3) 精读第 3、5 章，其中本科生可仅阅读第 3 章的边界值测试、等价类测试和场景法，以及第 5 章的静态白盒测试、对判定的测试、对路径的测试。略读第 4、6 章，在阅读的过程中应先自己尝试设计测试用例，然后对照书中内容，观察自己设计的测试用例与书上给出的测试用例有何异同。

(4) 结合自己参与开发的小型软件系统，阅读第 7～9 章，并注意针对系统中的不同模块进行测试阶段的实践，包括制订测试计划、设计测试用例和缺陷报告，其中测试用例报告、缺陷报告这类文档的撰写可参照第 10 章的内容。

(5) 阅读第 11 章，结合已经做过测试设计的小型软件系统，根据案例讨论中提到的测试设计思想和工具的使用方法，尝试设计和执行自动化单元测试、自动化功能测试以及性能测试。

本书配有电子课件、教学大纲、案例源代码或安装包、典型文档模板等教学资源，可登录电子工业出版社华信教育资源网(www.hxedu.com.cn)，免费注册、下载。

本书可作为高等学校计算机、软件工程专业“软件测试”课程的教材，也可作为软件测试人员的技术参考书。

参加本书编写工作的还有陈传波、肖来元、沈刚、薛志东。在本书的写作过程中得到了多方面的帮助、指导和支持。感谢华中科技大学的同事们，在业务交流中作者受益匪浅，特别感谢吴涛、区士颀、陆永忠、曾喻江老师，在共同的科研和教学实践活动中，为软件测试课程的规划与建设、课程实践案例的收集和整理、课程教学方法的改革探讨等方面，付出了大量辛勤的劳动；还要感谢学生们，在教学过程中，他们的想象力和创造力为本书的撰写带来了很多思想的火花，特别感谢胡媛媛、操智雄、程军、桂绍武、周俊全、董全立、贾春雨、严晓娥，他们为本书的案例整理付出了大量的心血；最后要感谢我的家人和朋友，没有他们默默的支持和鼓励，我无法完成这样一个艰巨的任务。

由于作者水平和时间所限，书中难免会出现一些错误和纰漏，请各界同仁不吝赐教。

编著者

目　录

第一部分　软件测试概述

第二部分　软件测试技术

第一部分　软件测试概述

第1章　软件测试核心概念

内容提要

本章将着重介绍与软件测试工程师关系最密切的核心概念：软件测试、软件缺陷、测试用例以及自动化测试，并以第二日问题为例，通过不断的测试尝试，帮助理解测试工作的内容和目标。

1.1　引子：猎人打鸟

在正式开始介绍软件测试之前，先来回答一个问题：如果树上有 10 只鸟，开枪打死 1 只，还剩几只？看看一个聪明学生的回答。

某日，老师在课堂上想考考学生们的智商，于是问一个学生："树上有 10 只鸟，开枪打死 1 只，还剩几只？"

学生反问："是无声手枪么？"　　老师："不是。"

学生："枪声有多大？"　　老师："80～100 分贝。"

学生："那就是说，枪声会震得耳朵发疼？"　　老师："是的。"

学生："在这个城市里打鸟犯不犯法？"　　老师："不犯。"

学生："您确定那只鸟真的被打死啦？"

"确定。"老师已经不耐烦了，"拜托，你告诉我还剩几只就行了，OK？"

学生："OK。鸟里有没有聋子？"　　老师："没有。"

学生："有没有关在笼子里的鸟？"　　老师："没有。"

学生："树的旁边还有没有其他的树，树上还有没有其他鸟？"　　老师："没有。"

学生："方圆 10 里呢？"　　老师："就这么一棵树！"

学生："有没有翅膀断了这种残疾的鸟？"　　老师："没有。"

学生："有没有饿的压根飞不动的鸟？"　　老师："没有，都身体倍棒。"

学生："算不算怀孕肚子里的小鸟？"　　老师："都是公的。"

学生："都不可能怀孕？"　　老师："@-@…，绝不可能。"

学生："打鸟人的眼花没花？保证是 10 只？"　　老师："没有花，就 10 只。"

老师脑门上的汗已经流下来了，下课铃也响了起来。

学生仍在发问，没有想停下来的意思："有没有傻得不怕死的？"　　老师："都怕死。"

学生："有没有对忠于爱情的，即使情侣被打中自己也留下来？"　　老师："我说了都是公的嘛！"

学生："同志可不可以啊！"　　老师："+_+…，性取向都正常！"

学生："会不会一枪打死 2 只？"　　老师："不会。"

学生："一枪打死 3 只呢？"　　老师："不会！"

学生："4只呢？"
学生："5只呢？"
学生："那6只总有可能吧？"
学生："...好吧，那么所有的鸟都可以自由活动么？"
学生："它们在受到惊吓而同时起飞的时候，会不会因为惊慌失措而互相乱撞上？"
老师："肯定不会！"
老师："绝对不会！！！"
老师："不——可——能！"
老师："完全可以。"
老师："不会，每只鸟都装有卫星导航系统，可以自由飞行。"
老师已经晕倒在地，这时只听学生满怀信心地回答："恩，如果您的回答没有骗人，打死的鸟要是挂在树上没掉下来，那么就只剩下一只鸟；如果那只打死的鸟掉下来了，树上就一只鸟都不剩了。"

如果你是一个软件测试工程师，你能像这个学生那样给出这样的回答吗？

这个故事是业界流行的一个笑话，说是笑话，其实也不完全是。故事中的学生可谓是一个优秀的软件测试人员。面对一个要交付给用户的软件产品，你也能够以这样挑剔的眼光，在有限的时间内找到软件中尽可能多、对软件破坏力最严重的那些缺陷吗？

下面先来看看与软件测试工程师密切相关的几个核心概念，初步体验一下测试工程师的工作吧。

1.2 软件测试的概念

1.2.1 软件的定义及特点

1. 软件的定义

软件与我们的生活息息相关，从微软的Windows XP操作系统、Office办公软件、腾讯的QQ聊天工具，到电视机、洗衣机等家用电器中内嵌的控制软件，我们每天都在使用着软件，那么，软件究竟是什么？

软件可表示成：

$$\text{软件(Software)} = \text{程序(P)} + \text{数据(库)(DB)} + \text{文档(D)} + \text{服务(S)}$$

其中，程序表示能够完成预定功能和性能的指令的集合，如C语言程序、Java程序等；数据(库)是依照某种数据模型组织起来并存储在二级存储器中的数据集合；文档指软件在开发、使用和维护过程中产生的文字与图形的集合，如《系统需求规格说明书》、《系统测试计划》、《用户手册》等；服务是指通过提供必要的手段和方法，满足接受服务的对象需求的过程，如安装指导、用户培训、售后技术支持、接受投诉等。随着计算机技术的飞速发展，特别是云计算的出现，其实质是通过提供云端技术服务来满足用户的需求，软件服务已不仅仅局限在通过人工来提供服务了，服务具有了更加广泛的内涵。

因此，软件测试不但包含传统意义上的针对可执行程序和源代码的测试，还应包括对数据、文档和服务的测试，软件测试工作应贯穿在整个软件的全生命周期中，在不同的软件开发阶段使用不同的测试策略，对应不同的测试内容，具有各自的侧重点。

2. 软件的特点

软件不同于硬件，软件是相对无形的东西，其特点如下：

(1) 软件必须依靠人的智力劳动才能创造出来，智力经验往往难以传承，且程序员在软件开发过程中常具有较大的随意性，使得开发的软件也常常具有较大的随意性。因此，软件测试人员必须具有异于常人的挑剔眼光，甚至使用一些旁门左道才可能发现潜伏的软件缺陷，导致大大增加了测试效果对软件测试人员经验的依赖性。

(2) 软件必须依托于具体的硬件设备才能运行，硬件的改变很可能导致软件不可用。因此，在软件测试工作中应针对各种主流的硬件设备支撑环境来测试软件是否可用。

(3) 软件不会如硬件一般产生磨损，但会随着其依托的硬件设备的变化，以及用户需求的不断变

化而需要进行升级，且到了某个时候，当需求和硬件的变化使得软件不得不改变其体系架构的时候，该软件就必须被淘汰而换之以全新的软件。因此，应测试升级后的软件对旧版本的兼容性。

1.2.2　软件测试的定义

顾名思义，软件测试(Software Testing)就是对软件的测试，需要通过什么方式和手段、针对软件中的哪些组成部分、按照怎样的流程来完成测试工作，以及其工作的最终目的是什么呢？围绕这些问题，人们对软件测试纷纷给出了自己的认识，在此不再详细给出。

IEEE 给出了关于软件测试的标准定义：软件测试是使用人工和自动手段来运行或测试某个系统的过程，其目的在于检验被测软件系统是否满足规定的需要，或是弄清楚被测系统的预期结果与实际结果之间的差别。该定义从 5 个方面体现了测试工作的核心与实质。

(1) 软件测试的根本目的是确保软件满足用户需求

“软件测试的目的在于检验被测软件系统是否满足规定的需要”，即保证被测软件符合用户需求是软件测试的最终目的，换句话说，只要软件系统不符合用户需求，就认为其存在缺陷，必须进行处理。因此，用户需求是测试的唯一依据，也是缺陷判定的根据。

案例

某公司(以下称客户)曾要求开发方为公司的重要客户(以下称用户)开发一套在安防中使用的视频监控系统的后台管理软件，当软件项目经理要求与软件使用方(即最终用户)见面来讨论系统需求的时候，客户却认为用户太忙，很难找到一个客户、用户及软件开发方可以同时到场的时间，且客户认为自己对于用户的需求是非常了解的，于是客户将需求定义工作包揽了下来，本应在用户与开发方之间的需求讨论变成了客户与开发方之间的讨论。软件从需求分析到测试完成，整个开发历时 3 个月。软件交付给客户后，正好面临客户投标，于是，客户将该软件拿到招标方去进行演示，不幸的事情发生了，招标方根本不认可这套软件，招标方认为，与其他的两个投标方的演示相比而言，虽然该投标方提供的硬件设备较为先进，但其配套提供的软件系统在用户的权限管理、监控管理等方面的功能存在较多不合理的地方，且与其他同类设备和系统的兼容性不够。导致整个软件系统需要做大量改动，甚至不如重做。

由此可以看出，无论是软件开发还是软件测试，都必须紧贴用户需求，凡是用户不认可的，都将视为缺陷。如果不能紧贴用户需求来做软件开发和测试工作，最终将导致项目失败。

另外，软件测试的最终目的是保证软件符合用户的需求，而非追求完美。图 1.1 给出了软件测试成本与收益之间的关系。从该图可以看出，随着时间的增长，通过测试发现的软件缺陷越来越多，在一段时间内将不断降低软件缺陷导致的损失，但随着时间的继续推移，软件中的主要严重缺陷已经发现并修复之后，继续执行软件测试只能发现一些对软件功能没有重大影响的轻微缺陷，且发现这些缺陷也变得越来越困难，反而导致测试成本的增长。因此，只要用户认为软件可以接受，就可以停止测试了。

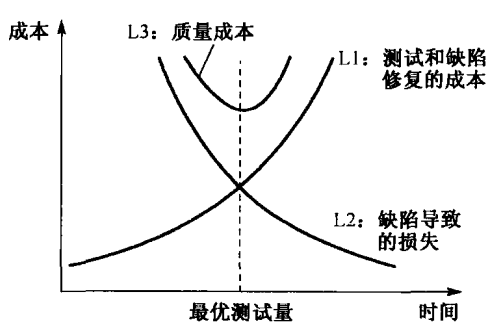

图 1.1　软件测试成本与缺陷导致损失的关系图

(2) 软件测试的目的是要衡量软件产品是否符合预期

尽管软件测试的根本目的是确保软件满足用户需求，但如何准确地理解和表达用户需求、如何验证软件是否符合用户需求呢？对于前者，显然是以系统需求规格说明书（以下简称 SRS）的形式来描述用户需求；对于后者，则需要通过设计测试用例(Test Case)，根据测试用例的执行情况来判

断其实际输出结果与预期结果之间是否存在差异，且这种差异是否超出用户可以接受的范围，若二者的差异用户可以接受，则无须理会，否则将对应一个软件缺陷，需对该缺陷进行适当的处理。当用户需求不多、不复杂的时候，例如，针对一个支持100个文档的全文检索工具，其测试工作并不困难，然而，若面对一个支持10000人同时在线、具备T级海量数据的智能搜索能力、且需与十几个已有系统进行接口的管理平台，就很可能面临上千个测试点交织在一起的复杂局面，如何高效地设计测试用例和实施测试，系统地衡量软件产品是否符合预期，是一项艰巨而富有挑战的任务。

(3) 软件测试是一个持续进行的过程

软件测试是持续进行的过程，不可能一蹴而就，必须重视软件测试过程管理。图1.2给出了软件测试的一般过程。图中实心箭头表示输入工作产品，空心箭头表示各测试阶段的输出物。

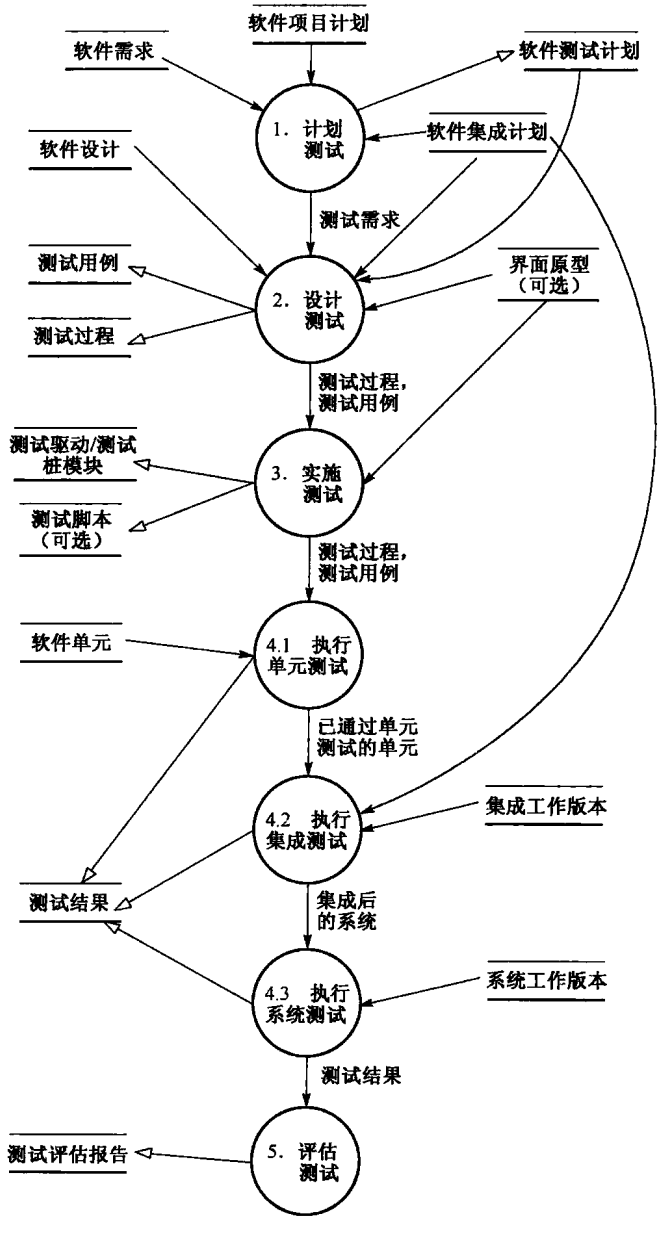

图1.2 软件测试的一般过程

从该图可以看出，软件测试的过程主要分为 5 个主要的步骤，即计划测试、设计测试、实施测试、执行测试和评估测试。具体说明如下。

① 计划测试。

计划测试是根据软件项目计划来制订测试计划，即在软件测试开始之前，必须清楚地回答一个问题：哪些人，在何时，需做哪些测试工作，使用怎样的方法，需要哪些资源，遵循怎样的规范来完成，可能碰到哪些风险，如何解决。具体说明如下。

哪些人：即与完成测试任务相关的人员，主要是测试团队的人员，也可能包含开发团队的人员和做市场的人员，应针对不同类型人员的特点和自身技术基础来布置不同的测试任务。

在何时：即制订进度表，根据项目开发计划来对应制订测试进度计划。

需做哪些测试工作：即为测试活动提供测试范围，应明确指出在某个测试阶段哪些测试需要做，哪些测试不需要做。例如，已发布过或已全面测试过的部分，已外包给其他公司开发实现的子系统，或者一些开源子系统等常常是不需要测试的。同时，应确定系统关键功能(如系统核心操作)和高风险功能，并划分优先级别。注意若在计划中列出某特性不需要测试，应明确指出该特性不测试的理由，否则易导致误解而使得本该测试的特性被遗漏。

使用怎样的方法：即测试策略，如使用的黑盒和白盒测试方法，是否需要设计桩和驱动模块，回归测试策略等。

需要哪些资源：即哪些测试需要借助某些硬件设备、软件工具来支持，如测试用例管理工具、缺陷管理工具、性能测试工具、测试仪等。

遵循怎样的规范：即在测试过程中必须严格遵守的规则，如进入和退出测试的标准、测试的暂停/恢复标准、测试阶段交付物的规定、相关测试文档模板等。

可能碰到哪些风险：即与测试相关的人员和资源风险，如人员的流失、资源的到位和使用风险等。不考虑风险就等于接受失败，必须对风险管理实行全局管理方法。

针对不同的测试阶段应分别制订各自的测试计划，如单元测试计划、集成测试计划、系统测试计划。

② 设计测试。

设计测试是对照测试计划中选定的测试范围，对测试计划中的方法进行细化。在测试设计阶段有一个重要的概念：测试需求，即需要测试的特性，一般地，通过分析功能需求，找到每个功能点所需要测试的特性，从而得到测试需求，再从测试需求设计测试用例。

同样地，针对不同的测试阶段将得到不同的测试设计文档，如单元测试设计说明书、集成测试设计说明书、系统测试设计说明书。测试需求和测试用例将包含在测试设计文档中。

③ 实施测试。

实施测试是测试过程中最重要的环节。测试设计阶段所设计的测试用例仅仅是以文档或管理工具等多种书面形式进行记录的，测试用例无法自动执行，且在一些特殊情况下，测试用例无法独立执行，因此，实施测试就是根据需要，通过脚本开发、测试驱动/测试桩模块的开发，将静态的测试用例转化为可以实际执行的动态测试用例。例如，当函数 f1 调用函数 f2，而函数 f2 又调用函数 f3 时，若需要对函数 f2 进行测试，但函数 f1、f3 尚未编码完成或尚未进行单元测试而无法确保其正确性，此时为了确保对函数 f2 实施单元测试，需要开发函数 f1、f3 对应的驱动和桩模块，使得从 f1 到 f3 的调用过程在有限条件下顺利执行，以达到测试用例运行的目的。

④ 执行测试。

执行测试是采用手动方式或通过测试脚本来运行测试用例的过程。在测试执行的过程中，需要注意记录测试过程以及在测试中发现的缺陷。在各个测试阶段都对应有执行测试的这个环节，但单元测试与集成测试是可以同步进行的，只有系统测试必须在单元测试和集成测试完成之后才能开始执行。

⑤ 评估测试。

测试执行过程中需要不断的评估，主要分为两方面工作，其一是对测试用例的执行结果进行评估，若测试用例的实际输出与预期输出相吻合，或二者之间的差异在用户可以接受的范围内，则测试用例的测试结果为通过，否则就认为用例的测试结果为失败，对应记录下相关的缺陷，该评估是在每次测试用例执行后必须做的事情；其二是对一段时间内的测试过程进行评估，这里所提的评估也主要是指阶段性的测试评估工作，该评估不仅包含对测试工作效果的评价，还包含对被测对象质量的评价，目的在于总结测试工作的经验、发现测试工作的不足，以及做出后续工作决策。例如，通过测试用例对代码行或功能点的覆盖、测试用例执行率指标等来判断测试工作是否全面、彻底，通过缺陷的个数、缺陷分布等指标来判断被测对象质量是否达到要求，若二者均满足要求，则认为当前的测试工作和被测对象质量达到测试的退出标准，可以进入后续的测试或开发环节，否则需继续进行测试。

(4) 测试既需要动态执行也需要静态检查

定义中提到的运行系统强调必须通过实际执行被测软件系统来发现缺陷，但并未提示该如何执行软件系统，测试人员的作用就在于通过设计测试用例，利用特定的数据集(即输入数据)和动作集(即软件操作步骤)来考验被测系统，判断系统能否以用户预期的方式给予正确、及时的反馈。同时，该定义还强调必须通过静态检查被测系统来发现缺陷，例如，对相关开发文档的评审、对源代码的走查、对用户手册等用户相关文档的检查，这一静态检查过程往往不需要实际运行程序，因此，在一定程度上将大大降低测试的工作量。

(5) 测试不仅需要手动执行还需要自动执行

传统的测试工作多依靠人工方式来完成，如测试计划的制订、测试用例的设计，甚至测试用例的执行也多依赖于人工执行，根据软件测试网在2011年所做的软件测试行业调查(以下称2011行业调查)可知，到目前为止，国内软件测试从业人员的工作类型主要集中在手动功能测试，所占比例为89%，其次是测试用例设计，所占比例为67%。然而，随着软件规模和复杂性的不断攀升，仅仅依靠手动来完成海量的测试工作显得越来越不现实，为了提高测试工作效率，加快测试工作进度，必须更多地引入自动化测试工具来辅助测试人员完成部分机械化的测试工作(如测试用例的执行、单元测试用例结果的检验、代码质量的度量等)，例如，2011行业调查表明，性能测试所占比例为35%，功能自动化测试比例为31%。

总之，从软件测试的标准定义可以看出，软件测试包含大量复杂的工作，要想将软件测试工作做好，需要解决至少如下3方面的问题：

① 围绕用户需求这个最根本的测试目的，需要考虑：如何有效地获取用户需求，如何准确理解和表达用户需求，如何保证用户需求的稳定性；

② 围绕软件产品是否符合预期这个测试目的，需要考虑：如何高效地设计测试用例，达到对成本、质量、进度的均衡控制；

③ 围绕测试过程的管理，需要考虑：如何合理评估和控制风险，如何规划整个测试工作，如何管理包括环境、工具、人力、测试交付物在内的所有相关资源。

后续章节将针对以上部分问题逐步展开详细地讨论。

1.2.3 捉虫实践1：很简单？

根据软件测试的定义可知，软件测试就是要满足需求，那么，只要了解了用户需求，不就可以进行测试了吗？利用第二日问题案例来亲自动手开始测试吧。

1．功能描述

第二日问题是一个简单的计算年月日的例子，其基本功能是：根据用户输入的有效日期(格式为年-月-日)，系统可以自动计算出下一天的日期。假如这就是用户需求，那么当不考虑如何接受用户输入，如何输出计算结果时，可将该需求细化为如下形式：

(1) 当用户输入为有效日期时，系统应自动计算出下一天的日期。其中，有效日期为 1800 年 1 月 1 日到 2050 年 12 月 31 日之间的所有日期，含 1800 年 1 月 1 日和 2050 年 12 月 31 日。

(2) 当用户输入无效时，即不满足有效输入条件时，系统不执行日期的计算，而是提示输入错误。

2．开始测试

根据需求描述，设计的测试如表 1.1 所示。

表 1.1　第二日的第一次捉虫实践：随机测试

输入数据(年-月-日)	预期输出(年-月-日)
1950-6-15	1950-6-16
1800-1-1	1800-1-2
2050-12-31	2051-1-1
2012-5-10	2012-5-11
2004-12-31	2005-1-1
2000-2-28	2000-2-29
1799-1-1	提示输入错误
2051-1-1	提示输入错误
2012-0-16	提示输入错误
2012-13-20	提示输入错误
2012-6-31	提示输入错误
2011-2-29	提示输入错误

特别提醒注意的是：针对 2050-12-31，其预期输出并不是提示错误，而是计算出其下一天的日期，其中原因请仔细阅读第二日问题的功能描述，应严格区分输入的有效域和输出的有效域。

3．测试分析

根据第二日问题细化后的需求，分别从有效日期和无效输入两方面着手，可以设计至少 12 个测试用例，那么，这就是测试吗？“很简单！我不需要任何计算机方面的知识，只要我有点常识，都可以设计这些用例啊。”当我们的嘴角浮现出轻蔑的微笑时，不妨深入思考如下问题。

Q：这些测试是如何设计得到的，是否存在一定的规律性？如果用别的日期来测试，能得到与这些数据一样的测试效果吗？

A：仅凭拍脑袋来测试是不行的，需要通过引入更多测试方法来指导测试的设计，以确保测试的效率。

Q：这些测试的质量如何？1800 年 1 月 1 日到 2050 年 12 月 31 日之间的日期那么多，为什么在此仅设计了 6 个测试，足够覆盖吗？在有效取值范围之外的日期那么多，也仅有 6 个测试，是否覆盖率太低？能不能直接选择有效范围内的所有日期进行穷举测试？

A：除了需要利用测试方法来提高测试效率之外，还需要引入测试标准，根据标准来衡量测试工作的充分性，控制测试的成本。