

人力资源和社会保障部软件测试师岗位培训指定教材

刘新生 主编

软件测试理论

Software Testing



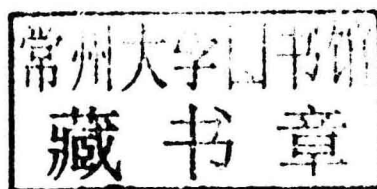
中国计量出版社

CHINA METROLOGY PUBLISHING HOUSE

人力资源和社会保障部软件测试师岗位培训指定教材

软件测试理论

刘新生 主编



中国计量出版社

图书在版编目 (CIP) 数据

软件测试理论/刘新生主编. —北京:中国计量出版社, 2010. 8

人力资源和社会保障部软件测试师岗位培训指定教材

ISBN 978 - 7 - 5026 - 3330 - 1

I. ①软… II. ①刘… III. ①软件—测试—工程技术人员—技术培训—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 158853 号

内 容 提 要

本书为人力资源和社会保障部软件测试师岗位培训指定教材。其内容包括: 软件测试基础、软件测试流程、软件测试技术、软件测试管理、软件测试工具。

本书除作为指定培训教材外, 还可作为高校软件测试等相关专业基础读本, 也可供相关培训机构和软件爱好者参考学习。

中国计量出版社出版

北京和平里西街甲 2 号

邮政编码 100013

电话 (010) 64215360

[http: //www. zgjl. com. cn](http://www.zgjl.com.cn)

北京市密东印刷有限公司印刷

新华书店北京发行所发行

版权所有 不得翻印

*

880 mm × 1230 mm 16 开本 印张 19.75 字数 547 千字

2010 年 8 月第 1 版 2010 年 8 月第 1 次印刷

*

印数 1—4 050 定价: 119.00 元

当前，信息化浪潮席卷全球，信息化程度已成为国力的重要标志，引起世界各国的高度重视。随着当代信息技术的迅速发展，实现可靠数据传送的计算机技术无疑是当代发展最为迅猛的科学技术之一。计算机的应用已经渗透到文化、科技、金融以及军事等领域，影响到我们生活的方方面面，几乎无处不在，成为人类改造自然不可或缺的重要工具。

蓬勃发展的计算机行业和不断提高的用户需求使得软件系统的规模和复杂性不断增加，随之而来的软件开发成本以及由于软件故障而造成的经济损失也不断增加，软件质量问题已成为人们共同关注的焦点。随着人们对软件本质的进一步认识，人们逐渐意识到软件工程和软件质量保证中存在的问题是制约软件产业快速发展的重要因素，软件测试对软件质量保证的作用已被大家认识并重视，软件测试在软件开发中的作用日益重要，软件测试的地位也得到了空前的提高。

随着软件规模及软件复杂度的日益增长，软件质量已成为不可忽视的问题。软件是一个逻辑体，软件中的错误都是由人类自己造成的。由于软件规模和软件复杂性等因素，使得证明软件是正确的非常困难。软件中的错误是不可避免的，人们只能根据需要尽可能地减少软件中的错误。软件（包括系统软件、应用软件及中间件等），尤其应用软件是非常关键的一环，它是应用系统的灵魂，体现应用系统的实质与特征。软件的质量与可靠性将对应用系统产生极大的影响，在一定程度上决定着应用系统的命运。

软件测试是保证软件质量的重要手段，它贯穿于软件开发的全过程，是软件生产与服务过程中的重要环节，在软件企业、IT服务企业、客户IT部门等机构中扮演着更为重要的角色。对于一个软件企业，“你不去发现缺陷，那一定是你的客户去发现缺陷”。提高软件产品质量已经成为增强企业竞争力的重要任务。软件测试的最终目的是“保证交付高质量的软件”。软件测试是目前验证软件是否达到了预期功能的最有效方法。作为软件产业的新兴门类，软件测试及其专业机构和组织发展迅速，软件测试人才需求旺盛，测试职业的价值也日益提升。近几年来，随着软件外包行业逐渐兴起和人们软件质量保障意识的加强，软件测试作为IT行业新的宠儿登上了历史舞台。

软件测试工作涉及诸多理论和实践，是一项需要专业技能、令人感兴趣并富有挑战性的工作。只有基于丰富的知识和经验，才能保证测试的深度和广度，进而保证测试的质量。应该说，要绝对保证软件的质量与可靠性是很困难的。首先，软件开发是带有“个人智慧与技巧”的性质；其次，千差万别和千

变万化且越来越复杂的应用需求,使得软件功能繁多,复杂而庞大;另外,软件开发方式和质量保证的技术与方法又滞后于需求的发展。

软件测试的重要性表现在两个方面。首先,软件测试工作直接关系到软件产品的质量,需要具有一定水平和实践能力的人员参与;其次,软件项目的测试成本要占整个软件产品开发成本的一半,甚至更多。随着我国软件产业的不断发展和规范化,对软件测试人员的需求也迅速地增长。因此,培养满足需求的、合格的软件测试人员是促进我国软件行业发展的重要任务。

近年来,随着社会对软件测试需求的增加,软件测试理论和技术得到了快速的发展。国际上一些著名的学术机构,以及微软、IBM等众多国际IT巨头的参与,使得软件测试理论正在逐步走向成熟,目前在全国已经形成近2000家的软件评测企业、数十万人的软件测试队伍。以软件测试工具、软件测试服务为主导的软件测试产业正在悄然兴起。

本书中,作者结合自己多年从业经验,以丰富充实的内容、活泼生动的文笔,阐述了软件测试行业的现状及软件测试技术基础知识等;此外,还列举了大家普遍关心的关于软件测试方面的问题以及作者的看法。相信这些内容对于那些想迈进软件测试大门却仍有顾虑,或者初学软件测试希望获得提高的朋友大有帮助。

希望本书能使更多具备相当计算机软件测试知识的学生能够投入到这一领域中来。“师傅领进门,修行在个人”。希望本书的读者不仅仅满足于本书的内容,要更系统地学习相关的大学计算机课程,才能拓展思维,更好地领会和实践。衷心希望这套丛书能够为培养一批有志于发展中国软件测试产业的测试技术人才和管理人才做出重要的贡献。

在本书编写过程中,得到了同行和朋友们的大力支持和帮助,他们在资料收集、整理和录入等工作中付出了辛勤劳动,在此表示衷心的感谢。书中不妥之处恳请读者批评指正。

编 者

2010年7月于北京

编 委 会

主 编 刘新生

副主编 潘 江 薛尧予 王巧霞

编 委 刘新生 潘 江 薛尧予 王巧霞 张 鹏

1 软件测试基础	1
1.1 软件与软件测试	1
1.1.1 什么是软件	1
1.1.2 软件测试的定义	3
1.1.3 软件测试的目的	3
1.1.4 软件测试的原则	4
1.1.5 软件测试的对象	6
1.1.6 软件测试的分类	7
1.1.7 软件测试的重要性	8
1.1.8 软件质量保证	9
1.1.9 软件测试行业的现状和前景	14
1.2 软件工程	17
1.2.1 什么是软件工程	17
1.2.2 软件工程的原则	18
1.2.3 软件工程的目标	20
1.2.4 软件生存周期	21
1.2.5 软件各阶段文档	25
1.3 软件开发	31
1.3.1 软件开发模型	31
1.3.2 软件开发流程	36
1.3.3 不同软件开发模式下的软件测试	37
1.4 数据库技术	40
1.4.1 数据库基本概念	40
1.4.2 数据库设计原则	42
1.4.3 数据库系统体系结构	43
1.4.4 数据库实例介绍	46
1.5 网络技术	49
1.5.1 计算机网络基础知识	49
1.5.2 ISO/OSI 参考模型	50
1.5.3 TCP/IP 参考模型	52

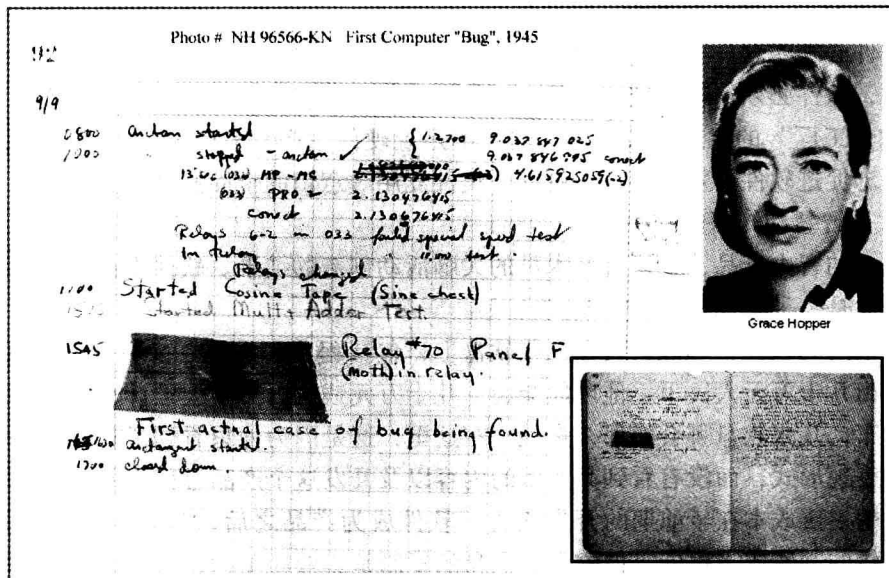
1.6 软件测试环境构造	54
1.6.1 软件测试环境介绍	54
1.6.2 虚拟机的使用	56
1.6.3 网络配置	57
1.6.4 服务器配置	59
1.6.5 系统备份与恢复	63
2 软件测试流程	66
2.1 软件测试模型	69
2.1.1 V 模型	70
2.1.2 W 模型	72
2.1.3 H 模型	74
2.1.4 X 模型	75
2.1.5 前置模型	76
2.1.6 软件测试模型的应用	76
2.2 软件测试阶段	78
2.2.1 概述	78
2.2.2 需求测试	79
2.2.3 单元测试	80
2.2.4 集成测试	85
2.2.5 系统测试	92
2.2.6 性能测试	95
2.2.7 用户测试	98
2.2.8 回归测试	101
3 软件测试技术	106
3.1 黑盒测试技术	106
3.1.1 概述	106
3.1.2 常用的黑盒测试方法	107
3.1.3 黑盒测试技术的应用	109
3.2 白盒测试技术	113
3.2.1 概述	113
3.2.2 白盒测试方法	114
3.2.3 面向对象的白盒测试	118
3.2.4 白盒测试综合策略	119
3.3 面向对象的软件测试技术	120
3.3.1 概述	120
3.3.2 面向对象测试模型	123
3.3.3 面向对象软件测试策略	127
3.4 应用负载压力测试技术	133
3.4.1 概述	133

3.4.2 负载压力测试策略	135
3.4.3 负载压力测试工具	136
3.4.4 负载压力测试技巧	139
3.5 Web 应用测试	140
3.5.1 概述	141
3.5.2 代码和组件测试	142
3.5.3 功能测试	143
3.5.4 性能测试	144
3.5.5 安全测试	148
3.6 网络测试	149
3.6.1 概述	149
3.6.2 网络仿真技术介绍	151
3.6.3 网络质量测试	156
3.6.4 网络应用测试	162
3.7 安全测试与评估	164
3.7.1 概述	164
3.7.2 安全系统测试策略	168
3.7.3 安全性测试方法	171
3.7.4 软件产品安全测试	173
3.8 兼容性测试	175
3.8.1 概述	175
3.8.2 硬件兼容性测试	175
3.8.3 软件兼容性测试	177
3.8.4 数据兼容性测试	181
3.8.5 平台兼容性测试	182
3.8.6 系统迁移测试	184
3.9 标准符合性测试	185
3.9.1 概述	185
3.9.2 标准符合性测试分类	187
3.9.3 标准符合性测试设计与实施	188
3.10 易用性测试	190
3.10.1 概述	190
3.10.2 安装易用性测试	190
3.10.3 功能易用性测试	192
3.10.4 界面易用性测试	193
3.11 可靠性测试	198
3.11.1 概述	198
3.11.2 可靠性测试用例设计	202
3.11.3 可靠性测试实施	203
3.11.4 可靠性评价	206
3.12 文档测试	209

3.12.1	概述	209
3.12.2	用户文档测试重点	211
3.12.3	测试用户手册	213
3.12.4	测试在线帮助	214
4	软件测试管理	216
4.1	软件配置管理	217
4.1.1	配置管理的概念	217
4.1.2	配置管理的目的	217
4.1.3	配置管理的功能	217
4.1.4	配置管理活动流程图	220
4.2	软件测试过程管理	222
4.3	软件测试需求管理	225
4.3.1	测试需求评审	225
4.3.2	测试需求跟踪	229
4.3.3	测试需求变更控制	235
4.3.4	测试需求检查	238
4.4	软件测试计划	240
4.4.1	测试方案的制定	240
4.4.2	制定测试的策略	244
4.4.3	测试计划编写及要素	247
4.5	软件测试用例设计	250
4.5.1	测试用例定义	250
4.5.2	测试用例的编写	251
4.5.3	测试用例跟踪和管理	256
4.6	软件测试执行	258
4.6.1	执行方式	258
4.6.2	执行流程	260
4.6.3	执行管理方法	261
4.7	软件缺陷管理	262
4.7.1	缺陷管理的目标	265
4.7.2	缺陷管理的一般流程	265
4.7.3	软件缺陷的编写	267
4.7.4	软件缺陷分析	270
4.8	软件测试风险分析	276
4.9	测试成本管理	285
4.9.1	总体指导原则	288
4.9.2	项目测试成本管理办法	288
4.9.3	产品测试成本管理办法	291
5	软件测试工具	292
5.1	软件测试工具介绍	292

5.1.1 软件测试自动化	292
5.1.2 软件测试工具分类	293
5.1.3 软件测试工具选择	295
5.2 Test Director 安装	296
5.2.1 Test Director 概述	296
5.2.2 安装前需知	297
5.2.3 安装步骤	298
5.3 创建项目	300
5.4 测试需求	300
5.5 测试计划	300
5.6 测试执行	301
5.7 缺陷跟踪	301
参考文献	303

1 软件测试基础



1.1 软件与软件测试

软件风险是普遍存在于任何软件开发项目中的实际问题。对于不同的项目，其差别只是风险有大有小而已。项目规模越大，问题越复杂，资源、成本、进度等因素的不确定性越大，承担项目所冒的风险也越大。风险是软件开发不可忽视的潜在不利因素，它可能在不同程度上损害到软件开发过程或软件产品的质量。软件风险驾驭的目标是在造成危害之前，及时对风险进行识别、分析，采取对策，进而消除或减少风险的损害。

1.1.1 什么是软件

“Software”这一名词于20世纪60年代初从国外传到国内，当时有人把它译为“软制品”，也有人将它译为“软体”，目前大多数人称之为“软件”。

国际标准化组织（ISO）将软件定义为：电子计算机程序及运用数据处理系统所必需的手续、规则和文件的总称。

对于上述定义，一种公认的解释是：软件是由程序、数据和文档这三部分内容组成的。即“软件 = 程序 + 数据（库） + 文档”。程序是为了取得一定的结果而编制的计算机指令的有序集合；数据是程序能进行正常加工信息所需要的原料，表现为一批符号的序列；文档是描述程序操作及使用的有关资料。

在软件的上述三种成分中，程序和数据是可由计算机执行的部分，而文档则是不能由计算机执行的部分。程序是软件的主体，因此在后面所说的“软件”主要是指程序。

软件是相对硬件而存在的。硬件是可以直观感觉到、触摸到的物理产品。生产硬件时，人的创造性的过程（设计、制作、测试）能够完全转换成物理的形式。例如，生产一个新的计算机，从

初始的草图、正式的设计图纸和面板的原型，一步一步演化成为一个物理的产品，如模具、集成电路、集成电路、电源和塑料机箱等。

软件则是逻辑的、知识性的产品集合，是对物理世界的一种抽象，或者是某种物理形态的虚拟化。

软件具有与硬件完全不同的特征，其主要表现在以下三个方面：

(1) 软件是核心，硬件是基础。

计算机硬件必须靠软件实现其功能，如果没有软件，硬件就好比一堆废铁，所以说软件是硬件的灵魂。同时，软件必须依赖于硬件，只有在特定的硬件环境上才能运行。

虽然“软件工厂”的概念也被引入，这并不是说硬件生产和软件开发是一回事，而是引用软件工厂这个概念促进软件开发中模块化设计、组件复用等意识的全面提升。

(2) 软件是智慧和知识的结晶。

软件是完全的智力产品，是通过技术员的大脑活动创造的结果。软件现在被认为属于高科技产品，而软件产业是一种知识密集型产业。

一个价值很高的软件，可能就装在几张软盘上，包括程序和文档。少数不了解软件价值的人，不愿意为此付出几十万元人民币。他可能会说，几十万元钱可以买一大堆计算机，可以买一辆桑塔纳或奥迪小轿车，几张软盘哪会值那么多钱？在这里，这是因为他只看到了软件的载体，也就是只看到其物理的表现形式，而没有看到其实质的内容以及开发这个产品过程中所投入的、高技术的大量人力。软件的主要成本在于前期的开发人力。软件成为产品之后，其后期维护、服务成本也很高。而软件载体的制作成本很低，如磁盘、光盘的复制是比较简单的，所以软件也就容易成为盗版的主要目标。

(3) 软件不会“磨损”，而是逐步完善的。

随着时间的推移，硬件构件会由于各种原因受到不同程度的磨损，但软件不会。新的硬件故障率很低，随着使用时间的延长，硬件会老化，故障率会越来越高。相反，隐藏的错误会引起程序在其生命初期具有较高的故障率，随着使用的不断深入，所发现的问题会慢慢地被改正，其结果是程序越来越完善，故障率会越来越低。

电脑软件一般分为系统软件和应用软件两大类。

系统软件是指管理和控制电脑资源的程序，是为使用电脑而提供的服务。常见的系统软件有：操作系统，如 DOS、Windows、UNIX 等；软件开发工具，如 Visual Basic、Visual C++、Power Builder 等；服务、实用工具程序，如 Norton Utilities、PCTools 等。

应用软件是面向各种应用领域的专用软件，是十分丰富且具有通用性的软件。常见的应用软件有：办公软件，如微软的 Office 套件、金山公司的 WPS 系列软件等；多媒体播放软件，如播放 VCD 的软解压程序等；图形处理软件，如 Paint Brush、Photoshop、Corel DRAW 等；网络浏览器，如 Internet Explorer、Netscape Navigator 等；教育软件，包括各种辅助英语、美术、电脑等的教育软件；游戏软件等。

仔细分析软件的组成，可以发现，实际上为了更为详细的描述软件，在程序、数据（库）、文档的基础上，还应加入另一项内容——服务。用一个简单的公式给出软件更为明确的定义：“软件 = 程序 + 数据（库） + 文档 + 服务”。

这是因为硬件和软件的维护差别很大。当一个硬件构件磨损时，可以用另外一个备用零件替换它，但对于软件，不存在替换，而是通过开发补丁程序等服务手段不断地解决其适用性问题，或扩充其功能。一般来说，软件维护要比硬件维护复杂得多，而且软件的维护周期要长得多。软件正是通过不断的维护，改善功能，增加新功能，来提高软件系统的稳定性和可靠性的。

1.1.2 软件测试的定义

随着信息技术的飞速发展，使软件产品应用到社会的各个领域，软件产品的质量逐渐成为人们共同关注的焦点。软件的生产者和使用者均面临严峻的考验，软件开发商为了占有市场，必须把产品质量作为企业的重要目标之一，以免在激烈的竞争中被淘汰出局。

用户则希望选用优质的软件，以保证自己业务的顺利完成。质量不佳的软件产品会大大增加开发商的维护费用和用户的使用成本，还会产生其他的责任风险。在一些关键应用（如民航订票系统、银行结算系统、证券交易系统、自动飞行控制软件、军事防御和核电站安全控制系统等）中使用质量有问题的软件，会造成灾难性的后果。

软件危机曾经是软件界甚至整个计算机界最热门的话题。现在人们已经逐步认识到软件危机实际上仅是一种状况，那就是软件中有错误，正是这些错误导致了软件开发在成本、进度和质量上的失控。

通常称之为 Bug 的软件缺陷是伴随着软件出现的，随着软件缺陷（Bug）日益增多，造成了日益严重的质量事故。因此，人们“对抗”软件缺陷（Bug）的态度日益坚决，使得软件测试不断地得到加强、重视和持续发展。

第一个软件缺陷（Bug）发生在 1945 年 9 月的某一天，在一间老式建筑的窗户外面飞进来一只飞蛾，G. Hopper 正埋头工作在一台名为 Mark H 的计算机前，这台计算机使用了大量的继电器（电子机械装置，那时还没有使用晶体管），突然 Mark H 死机了。G. Hopper 试了很多次还是不能启动，G. Hopper 开始用各种方法查找问题，最后确定是某个电路板的继电器出错了，观察这个出错的继电器，G. Hopper 惊奇地发现一只飞蛾躺在里面。由此，Hopper 在他的“事件记录本”中，写上了“第一个发现虫子的实例”。

软件缺陷这一概念用来描述各种软件错误，是所有软件错误的统称。把符合下列 5 种特征之一的软件错误认为是软件缺陷：

- （1）软件未达到软件产品需求说明书中指明的要求；
- （2）软件出现了软件产品需求说明书中指明不会出现的错误；
- （3）软件功能超出了软件产品需求说明书中指明的范围；
- （4）软件未达到软件产品需求说明书中虽未指明但应达到的要求；
- （5）测试人员认为难以理解、不易使用、运行速度缓慢或者最终用户认为不好的问题。

如何去避免错误的产生和消除已经产生的错误，使程序中的错误密度达到尽可能低的程度，是软件测试工作的主要内容。

测试的英文单词为“test”，即检验或考试之意。所谓测试，就是通过一定的方法或工具，对被测试对象进行检验或考试，目的是发现被测试对象存在的问题。软件测试是测试中的特例，它的测试对象是人类的智力产品——软件。因此，软件测试是所有测试中最复杂的一种测试。

1983 年，IEEE 将软件测试定义为：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，或弄清预期结果与实际结果之间的差别。可以看出，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审，是软件质量保证的关键步骤。

大量统计资料表明，软件测试的工作量往往占软件开发总工作量的 40% 以上。极端情况下，测试关系人的生命安全的软件所花费的成本，可能相当于软件工程其他开发步骤总成本的三倍到五倍。因此，必须高度重视软件测试工作，绝不要以为写出程序之后软件开发工作就接近完成了，实际上大约还有同样多的开发工作量需要完成。

1.1.3 软件测试的目的

G. J. Myers 在其《软件测试技巧》一书中有如下的论述：“程序测试是为了发现程序中的错误

而执行程序的过程；好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案；成功的测试是发现了迄今为止尚未发现的错误的测试。”或者说，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（即输入一些数据而得到其预期的结果），并利用这些测试用例去运行程序，以发现程序错误的过程。

仅就测试而言，其目标是发现软件中的错误。但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。分析错误产生的原因和错误的分布特征，可以帮助我们发现当前所采用的软件过程的缺陷并加以改进。同时，这种分析也能帮助我们设计出有针对性的检测方法，改善测试的有效性。

理想情况下，软件测试就是为了找出被测试软件的所有错误。事实上，我们不可能发现所有的问题，只能尽力地暴露软件存在的问题，及时发现软件中潜藏的错误，帮助设计人员完善软件。

如果我们的测试过程运行多次而未发现软件错误，这样一定程度上可以得出：被测试软件已经完美了，或是需要重新构建一套测试过程。因为存在针对性，所以软件测试存在多种目的，可归结为最重要的三条：

（1）使软件满足客户的需求。软件工程的根本目标是开发出高质量的完全符合用户需要的软件。不管是客户定制软件还是开发软件产品，都是为了满足客户的需求。从用户的角度出发，普遍希望通过软件测试暴露出软件中隐藏的错误和缺陷，以考虑是否可以接受该产品。以定制软件为例，需求报告中双方确认的软件功能、性能和接口，就是客户验收测试（Beta 测试）的依据。

（2）便于控制开发团队的工作状态和工作进度。计划质量管理部门，只有通过文档或程序的测试，才能掌握开发团队真正的工作状态和工作进度，避免犯主观主义和官僚主义的错误。

（3）便于控制版本的发布。对市场发布新版本，事先必须通过单元测试、集成测试、Alpha 测试和 Beta 测试，形成测试文档和测试管理文档，甚至建立测试数据库。只有这些工作都做完了，才达到了预期的要求，新版本才能对外发布。

由于测试的目标是暴露程序中的错误，由程序的编写者自己进行测试一般来说是不恰当的。对于长度仅有数百行的小程序，测试和纠错可由编码者一人完成，但对于大型的程序而言，测试与纠错必须分开进行。为了保证大程序的测试不受干扰，通常都把它交给独立的小组进行，等发现了程序有错误，再退回编码者进行纠错。

此外，应该认识到测试决不能证明程序是正确的。即使经过了最严格的测试之后，仍然可能还有没被发现的错误潜藏在程序中。测试只能查找出程序中的错误，不能证明程序中没有错误。

1.1.4 软件测试的原则

软件测试的基本原则是站在用户的角度，对产品进行全面测试，尽早、尽可能多地发现缺陷（Bug），并负责跟踪和分析产品中的问题，对不足之处提出质疑和改进意见，“零缺陷（zero - Bug）”是一种理想，“足够好（good - enough）”是测试的原则。

在软件测试过程中，应注意和遵循以下原则：

（1）所有的测试都应该追溯到用户需求，致力于满足用户需求。

所有测试的指标都应建立在用户需求之上，因为软件测试的目标在于揭示错误，所以测试人员要始终站在用户的角度去看问题，找出导致程序不能满足用户需求的严重错误。软件测试必须基于“质量第一”的思想去开展各项工作。当时间和质量冲突时，时间要服从质量。

（2）尽早制定测试计划。在整个开发过程中要“尽早地和不断地进行软件”测试。

由于用户需求的复杂性和不确定性、软件的复杂性和抽象性、软件开发各阶段工作的多样性，加上软件开发各种人员的配合差异等因素，使得开发的每个环节都可能产生错误。

所以，测试应贯穿到整个软件开发过程，不应该只简单地看作软件开发的一个独立阶段。软件项目一启动，软件测试也就开始，而不是等程序写完，才开始进行测试。软件开发过程中要尽早

地、不断地发现错误，及时纠正，提高软件质量。

软件测试计划是做好软件测试工作的前提。所以在进行实际测试之前，应制定良好的、切实可行的测试计划并严格执行，特别要确定测试策略和测试目标。

测试计划应包括被测试软件的功能、输入和输出、测试内容、各项测试进度安排、资源要求、测试资料、测试工具、测试用例、测试控制方式和过程、系统组装方式、跟踪规程以及回归测试的规定和评价标准等。测试计划可以在需求模型一完成就开始，详细的测试用例定义可以在设计模型被确定后开始。值得注意的是，应该长期保存测试计划，直到软件系统被废弃为止。

(3) 应事先定义好产品的质量标准。

基于给定的产品质量标准，通过分析相应的测试的结果，对产品的质量进行分析和评估，做到“有的放矢”。

(4) 测试工作不应由系统开发人员或开发机构本身来承担，应交由第三方进行。

从心理学角度讲，人们往往不愿意否定自己的工作。此外，测试是带有“挑剔性”的行为，即如果程序中包含了由于程序员对程序功能的误解而产生的错误，当程序员测试自己的程序时，往往还会带着同样的误解而使错误难以发现。相对而言，第三方进行测试会更客观，更有效，为了得到最好的测试效果，应由第三方人员对软件进行客观、严格的独立测试。

(5) 穷举测试是不可能的。

考虑到程序路径排列的数量巨大，因此测试过程中，不可能输入一切可能的数据，让程序都执行一遍。应采用选择测试，即选取能充分覆盖程序逻辑，并确保程序设计中使用的所有条件是有可能和有代表性的，并以典型的数据作为测试用例。

除了检查程序是否做了它应该做的事，还要看程序是否做了它不该做的事。因此，除了选用合理条件的输入数据，还要选用不合理条件的输入数据作为测试用例的数据。过去的程序测试中，考虑比较多的是合法的、合理的输入条件的测试，用以检查程序是否能正确完成它的功能，而忽视了不合法的和预想不到的输入条件。然而，用户在使用软件的过程中，往往会出现不遵循事先约定的情况。例如，在键盘上按错了键，输入非法命令及不合理的数据等。如软件中缺乏相应的处理功能，那么就容易产生故障，甚至导致软件失效。软件测试过程中，须用不合理的输入条件测试程序，进而检验软件系统处理非法命令的能力。

测试的关键技术是设计一组高产的测试用例，好的测试方案是尽可能发现至今为止仍未发现的错误。测试用例是设计出来的，不是写出来的，所以要根据测试的目的，采用相应的方法去设计测试用例，从而提高测试的效率，更多地发现错误，提高程序的可靠性。从某种意义上说，测试是否成功，取决于测试用例的选择。所以测试用例的正确性和合理性具有很重要的意义。

(6) 设计测试用例时，要给出测试的预期结果。

这条原则是以心理学为基础的，如果事先无法肯定预期的测试结果，由于受“眼睛会看见事先想看见的东西”现象的影响，往往会把看起来似是而非的东西当成是正确的结果。解决这个问题基本的方法是，事先给出程序预期的输出结果，并以此为标准详细检查所有的输出，抓住症状并揭示错误。因此一个测试用例必须包括两部分：对程序输入数据的描述和由这些输入数据应产生的输出结果的精确描述。

(7) 对合理的和不合理的输入数据都要进行测试。

为了提高程序的可靠性，不仅要考虑合理的输入数据，同时也应考虑不合理的输入数据。合理的输入数据可以用来验证程序的正确性，而不合理的输入数据是指异常的、临界的、可能引起问题异变的输入数据。但在测试程序时，人们很容易将注意力集中在合理的和预期的输入情况上，而忽视不合理和非预期的情况。事实上，软件在投入运行后，用户往往会不遵循合理的输入要求，而进行了一些非法的输入，如果系统不能对此意外输入做出正确反应，系统将很容易产生故障，甚至造

成系统的瘫痪。因此在测试时必须重点测试系统处理非法输入和命令的能力，而且用不合理的输入数据进行测试往往比用合理的输入数据进行测试能发现更多的错误。

(8) 测试应从“小规模”开始，逐步转向“大规模”。

最初的测试通常把焦点放在单个程序模块上，进一步测试的焦点则转向在集成的模块簇中寻找错误，最后在整个系统中寻找错误。随着测试的逐步深入展开，要集中测试容易出错的地方。

(9) 充分注意到错误的群集现象。

统计结果表明，测试发现的错误中有 80% 的错误很可能是由 20% 的程序模块造成的，即程序中存在错误群集性现象。也就是说，程序段中错误数目多的地方，残存错误的数目也比较多。这和程序员的编程水平和习惯有很大的关系。因此，为了提高测试的效率，在进行深入测试时，应对错误群集的程序区段进行重点测试。

(10) 测试过程中，应重视并妥善保存文档。

应妥善保存一切测试过程文档。包括测试计划、测试用例、测试报告等。这些都是检查整个开发过程的主要依据，在对该软件产品进行维护时，十分需要这些测试文件，以便修改后再测试。同时，这些文档也是测试人员的智慧结晶和经验积累，对新人或今后的工作都有指导意义。

原则是重要的，方法应该在这个原则指导下进行。除了上述原则之外，测试过程中，还要注意以下事项。

(1) 要注意回归测试的关联性问题，修改一个错误而引起更多错误出现的现象普遍存在。

(2) 在规划测试时，不要设想程序中不会查出错误，默认程序中不存在错误对测试工作极为不利。

(3) 除检查程序功能是否完备外，还要检查程序功能是否有多余。

(4) 对做了修改之后的程序进行重新测试时，应严格执行测试用例，否则将有可能忽略由修改错误而引起的新错误。

(5) 必须彻底检查每一个测试结果，保证所有的测试结果无一遗漏。

(6) 对测试错误结果一定要有一个确认的过程。一般由“甲”测试出来的错误，一定要有一个“乙”来确认。严重的错误可以召开评审会进行讨论和分析。

1.1.5 软件测试的对象

统计资料表明，在查找出的软件错误中，需求分析和软件设计的错误约占 64%，程序编码错误约占 36%。这表明，对程序而言，其许多错误都是因为前面阶段工作中所产生的问题未及时解决而引起的。在对“需求理解与表达的正确性、设计与表达的正确性、实现的正确性以及运行正确性”的验证中，任何一个环节发生的问题都可能在软件测试中表现出来。

因此，软件测试不仅仅是源程序的测试，也就是说，软件测试不等于程序测试。需求分析、概要设计、详细设计，以及程序编码等各个阶段所得到的一切文档和程序，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。

必须坚持阶段评审，以确认和验证各阶段的工作。确认与验证工作均属于软件测试。

确认的目的是验证在给定的外部环境中软件的逻辑正确性，它包括需求规格说明的确认和程序的确认，程序的确认又分为静态确认与动态确认。静态确认一般不在计算机上实际执行程序，而是通过人工分析或程序正确性证明来确认程序的正确性，也就是通常所说的代码走读（code review）。动态确认主要是通过动态分析和程序测试来检查程序的执行状态，以确认程序的正确性。

验证是试图证明在软件生存期中的各个阶段及其阶段之间的逻辑协调性、完备性和正确性。图 1.1 所描述的就是软件生存期各个重要阶段之间所要保持的各类正确性问题。它们是验证的主要对象，包括：用户要求、需求说明书、设计说明书、源程序及运行结果。