



全国信息技术人才培养工程指定培训教材



全国信息技术水平考试指定教材



软件测试工程师高级职业教育系列教程

软件测试 理论与实践教程



张向宏 主编 陈淦萍 副主编

工业和信息化部电子教育与考试中心 组编



人民邮电出版社
POSTS & TELECOM PRESS



全国信息技术人才培养工程指定培训教材



全国信息技术水平考试指定教材



软件测试工程师高级职业教育系列教程

软件测试 理论与实践教程

张向宏 主编 陈淦萍 副主编

工业和信息化部电子教育与考试中心 组编

人民邮电出版社

北京

图书在版编目 (C I P) 数据

软件测试理论与实践教程 / 张向宏主编; 工业和信息化部电子教育与考试中心组编. —北京: 人民邮电出版社, 2009. 6
ISBN 978-7-115-19923-2

I. 软… II. ①张…②工… III. 软件—测试—教材
IV. TP311.5

中国版本图书馆CIP数据核字 (2009) 第053341号

内 容 提 要

本书全面、系统地介绍了软件测试的理论知识与常用方法, 主要内容有软件测试的基础知识、软件测试的标准、白盒测试技术、黑盒测试技术、性能测试技术、软件测试案例分析, 以及软件测试的前沿技术。本书附录中简要介绍了一些常用的测试工具。

本书既重视软件测试理论知识的介绍, 亦重视软件测试实际操作方法的讲解, 能够帮助读者了解软件测试工程师的职业要求, 熟悉软件测试流程, 掌握软件测试的基本思路, 并能运用一些常用工具进行软件测试工作。

本书既可以作为软件测试工程师培训项目与各类院校软件相关专业的教学用书, 也可以作为软件测试的爱好者以及软件公司测试人员的自学教程。

软件测试理论与实践教程

- ◆ 主 编 张向宏
副 主 编 陈淦萍
组 编 工业和信息化部电子教育与考试中心
责任编辑 李 莎
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 14.5
字数: 354 千字
印数: 1—4 000 册
- 2009 年 6 月第 1 版
2009 年 6 月河北第 1 次印刷

ISBN 978-7-115-19923-2/TP

定价: 32.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

序

软件产业发展的几十年中，软件测试已逐步渗透到各个领域，成为越来越不可缺少的技术成分。例如，国家认定软件作为工业和信息两化融合的纽带，使信息化带动工业化、工业化促进信息化，将软件提高到了一个很重要的地位。由此看来，软件的品质已成为人们日益关注的重中之重。如何找到一种全面的分析方法检测软件开发过程中不同阶段的结果，以便尽可能早地、系统地保证或提高软件产品的质量和可靠性，从而减少后期“弯腰”的必要性与次数，已成为影响软件企业生产力与生产效率的关键问题。

目前，越来越多的软件公司和管理技术人员在工作中将更多的时间和资源投向了测试方面。很多优秀企业中，开发人员与测试人员的比例达到了 3:1 或 2:1，许多顶尖的技术人员在从事质量控制和软件测试工作。而国内这几年软件测试人员的严重短缺和招聘难度的加大从反面证明了软件测试正越来越得到重视。

为此，国家不断地制定政策措施，促进软件产量的提高和软件质量的提升，加速人才队伍的建设。2000 年，国务院颁发 18 号文件《关于鼓励软件产业和集成电路产业发展的若干政策》，特别强调“积极支持企业、科研院所和社会力量开展各种软件技术培训，加强在职员工的知识更新与再教育”。2002 年，国务院办公厅颁发 49 号文件《振兴软件产业行动纲要》，再次强调“面向企业和市场需求，通过学历教育、职业教育、继续教育和培训等多种形式，加快培养软件经营管理人才、国际市场开拓人才、精通行业应用的高级软件人才，改善软件人才结构”。2004 年，科技部颁发 124 号文件《关于进一步提高我国软件企业技术创新能力的实施意见》，更是明确强调“加强评测工作”以及“加快培养软件测试人才”。

中国的大学很少开设软件测试专业，软件测试人才培养体系也很不健全，且培养模式单一、课程相对落后。这导致培养的人才最终不具备企业所需的专业技能，无法满足企业的用人需求。为贯彻落实有关信息技术人才培养的计划，针对软件测试行业现在面临职业人才匮乏的问题，特编写了这本软件测试工程师职业培训教材。本教材既详细阐述了软件测试理论知识，又介绍了软件测试的实战经验，对任何从事软件测试和软件开发的人员，以及软件工程相关专业的高校师生，都具有重要的参考价值。希望本教材对广大读者有所裨益，能为行业尽快培养大批职业化的软件测试专业技术人才做出贡献，以保障软件产业持续、健康发展。

工业和信息化部软件服务业司司长 赵小凡

2009 年 1 月

前 言

本教材较为全面地涵盖了软件测试专业的知识，追溯了软件测试的发展史，反映了当前国际上采用的最先进的测试理论、标准、技术和工具，展望了软件测试的发展趋势，强调了表述的准确性、知识的系统性以及技术的先进性和实用性。软件测试是理论与实践紧密结合的工作，为了具有较强的实践性，本教材提供了功能测试、白盒测试、性能测试的示范测试案例，帮助读者熟悉软件测试工程师职业的基本要求，掌握分析问题的基本思路和基本方法，进一步提高在实际工作中解决问题的能力。本教材还就测试的前沿技术进行了一些探讨，包括敏捷测试技术和测试驱动开发技术，使读者能够了解目前的先进技术，有利于进一步提高。

本教材在编写过程中，十分重视理论联系实际，在内容上做到深入浅出，通俗易懂，既适用于课堂教学，又能满足读者自学的需求。本教材适合软件测试培训机构和软件学院使用，也适合当前软件开发企业项目经理、软件测试专业人员、想了解软件测试的软件开发人员，以及想进入软件测试领域的新手学习。

本教材由张向宏担任主编，陈涿萍担任副主编，编写提纲由陈涿萍拟定。其中，第 1 章、第 8 章由陈涿萍编写，第 2 章由罗文兵、郭宇编写，第 3 章由唐刚编写，第 4 章由陶新昕编写，第 5 章由吴妮妮、唐磊、李艳波编写，第 6 章由郭盈、王博洋编写，第 7 章由唐磊、陶新昕、郭盈编写。全书由李艳波、郭盈、陈涿萍统稿。

在本书的编写过程中，多次得到了工业和信息化部电子教育与考试中心高级工程师盛晨媛的指导，促使我们不断地改进书中的内容及形式，另外还得到了相关领导的大力支持，在此，向他们表示衷心的感谢！

由于时间有限，本教材中难免有一些缺点和不足之处，欢迎广大读者批评指正。我们的联系邮箱是 lisha@ptpress.com.cn。

编 者
2009 年 1 月

目 录

第 1 章

软件测试概述 1

- 1.1 软件测试的起源及发展 2
- 1.2 软件测试的现状 & 前景 4
- 1.3 软件测试工程师应具备的素质 5

第 2 章

软件测试基础 7

- 2.1 软件测试的基本概念 8
 - 2.1.1 软件生命周期 8
 - 2.1.2 软件测试的定义 9
 - 2.1.3 软件缺陷 11
 - 2.1.4 软件测试的目的 12
 - 2.1.5 软件测试的原则 12
- 2.2 软件测试的类型 14
 - 2.2.1 按照开发阶段划分 14
 - 2.2.2 按照测试实施组织划分 15
 - 2.2.3 按照测试技术划分 16
 - 2.2.4 按照测试内容划分 17
- 2.3 软件测试过程模型 21
 - 2.3.1 V 模型 22
 - 2.3.2 W 模型 23
 - 2.3.3 H 模型 24
 - 2.3.4 X 模型 25
 - 2.3.5 前置测试模型 26
- 2.4 软件质量保证 28
 - 2.4.1 软件质量保证的定义 28
 - 2.4.2 软件质量保证的工作内容 29
 - 2.4.3 软件质量保证的工作过程 30
 - 2.4.4 软件质量保证与软件测试的区别 31

第 3 章

软件测试标准 33

- 3.1 软件质量与评价 34
 - 3.1.1 软件质量与度量 34

- 3.1.2 软件质量模型 34
- 3.1.3 软件质量评价 37

- 3.2 软件测试标准 38
 - 3.2.1 国内外软件测试标准概述 38
 - 3.2.2 软件测试通用标准 39
 - 3.2.3 军用软件测试标准 54
 - 3.2.4 信息系统安全评估标准 55
- 3.3 软件测试标准的行业应用 60

第 4 章

白盒测试技术 62

- 4.1 白盒测试概述 63
 - 4.1.1 词法分析与语法分析 63
 - 4.1.2 静态错误分析 64
 - 4.1.3 程序插桩技术 64
- 4.2 静态测试 69
 - 4.2.1 代码检查法 69
 - 4.2.2 静态结构分析法 81
 - 4.2.3 静态质量度量法 84
- 4.3 动态测试技术 88
 - 4.3.1 逻辑覆盖法 88
 - 4.3.2 基本路径测试法 93
- 4.4 白盒测试综合策略 98
 - 4.4.1 最少测试用例数的计算 98
 - 4.4.2 测试覆盖准则 100

第 5 章

黑盒测试技术 103

- 5.1 黑盒测试概述 104
 - 5.1.1 黑盒测试的概念 104
 - 5.1.2 黑盒测试的特点 104
 - 5.1.3 黑盒测试的应用 105
- 5.2 功能测试用例设计方法 105
 - 5.2.1 等价类划分法 105
 - 5.2.2 边界值分析法 107
 - 5.2.3 场景法 108

5.2.4	因果图法	110
5.2.5	正交实验法	112
5.2.6	判定表法	116
5.2.7	其他方法	118
5.2.8	测试方法综合使用策略	119
5.3	黑盒测试用例的组织和编写	119
5.3.1	编写测试用例的意义	119
5.3.2	如何编写有效的测试用例	120
5.3.3	测试用例的编写原则	121
5.3.4	测试用例的构成	123
5.3.5	测试用例的编写策略	124
5.4	QTP 自动化工具的使用	124
5.4.1	QTP 概述	125
5.4.2	QTP 的工作原理及适用范围	125
5.4.3	QTP 窗口结构	126
5.4.4	QTP 工作流程	128
5.4.5	录制测试脚本	128
5.4.6	增强测试脚本	130
5.4.7	执行并分析测试脚本	137

第6章

性能测试技术 140

6.1	性能测试概述	141
6.1.1	软件性能与性能测试	141
6.1.2	性能测试的目的	141
6.1.3	性能测试的常用术语	143
6.2	性能测试的分类	145
6.2.1	主要类型	145
6.2.2	其他类型	147
6.3	性能测试的策略及实施流程	148
6.3.1	应用在客户端的性能测试	148
6.3.2	应用在网络的性能测试	149
6.3.3	应用在服务器端的性能测试	150
6.3.4	性能测试的实施流程	151
6.4	性能测试实战技术	153
6.4.1	简介性能测试的常用工具	153
6.4.2	结合 LoadRunner 实施性能测试	154
6.4.3	开发测试脚本	156
6.4.4	执行测试	163
6.4.5	分析测试结果	165

第7章

测试案例分析 166

7.1	功能测试案例分析	167
7.1.1	“122 接处警”系统简介	167
7.1.2	前期分析	167
7.1.3	用例设计方法的分析	170
7.1.4	测试用例编写策略的具体应用	175
7.2	白盒测试案例分析	175
7.2.1	静态测试	176
7.2.2	覆盖率测试	182
7.3	性能测试案例分析	194
7.3.1	办公自动化系统的案例分析	194
7.3.2	workflow 引擎的案例分析	196

第8章

测试前沿技术 200

8.1	敏捷测试技术	201
8.1.1	敏捷方法的特征	201
8.1.2	敏捷方法的质量	201
8.1.3	敏捷测试的引入	202
8.1.4	敏捷测试用例设计	202
8.1.5	敏捷测试的弱点	204
8.2	测试驱动开发 (TDD)	205
8.2.1	TDD 的优势	205
8.2.2	TDD 的原理	206
8.2.3	TDD 的过程	206
8.2.4	TDD 的原则	207
8.2.5	TDD 的测试技术	208
8.2.6	TDD 测试案例	209

附录

测试工具介绍 211

A.1	配置/过程管理工具	212
A.2	功能测试工具	215
A.3	性能测试工具(系统强度测试工具)	216
A.4	白盒、嵌入式测试工具	218
A.5	软件开发工具	221
A.6	仪器仪表	223
A.7	其他工具	224

第1章 软件测试概述

本章阐述了软件测试的起源及发展历程，分析了软件测试的现状及前景，特别是我国软件测试的现状与发展前景，介绍了软件测试工程师应具备的素质，以使读者对软件测试行业有个整体的了解。

1.1 软件测试的起源及发展

软件测试是伴随着软件的产生而产生的。在早期的软件开发过程中，软件规模小、复杂程度低，软件开发的过程混乱无序、相当随意。人们对测试的理解也比较狭窄，开发人员将测试等同于“调试”，目的是纠正软件中存在的故障，通常由开发人员自己完成这部分工作。当时对测试的投入极少，并且测试介入也晚，常常是等到形成代码，产品已经基本完成时才进行测试。

直到 1957 年，软件测试才开始与调试区别开来，并作为一种发现软件缺陷的活动。由于一直存在着“为了让我们看到产品在工作，就得将测试工作往后推一点”的思想，在潜意识里对测试的目的就理解为“使自己确信产品能工作”。测试活动始终落后于开发活动，测试通常被作为软件生命周期中最后一项活动而进行。当时也缺乏有效的测试方法，主要依靠“错误推测 (Error Guessing)”来寻找软件中的缺陷。因此，大量软件在交付后仍存在很多问题，软件产品的质量得不到保证。

到了 20 世纪 70 年代，这个阶段开发的软件仍然并不复杂，但人们已开始思考软件开发流程的问题，尽管对“软件测试”的真正含义还缺乏共识，但这一词条已经频繁出现，一些软件测试的探索者们建议在软件生命周期的开始阶段就根据需求制订测试计划。这时也涌现出一批软件测试的宗师，Bill Hetzel 博士就是其中的领导者。1972 年，软件测试领域的先驱 Bill Hetzel 博士（代表论著《The Complete Guide to Software Testing》）在美国的北卡罗来纳大学组织了历史上第一次正式的关于软件测试的会议。在 1973 年，他首先给了软件测试一个这样的定义“就是建立一种信心，认为程序能够按预期的设想运行 (Establish confidence that a program does what it is supposed to do)”。后来在 1983 年他又将定义修订为“评价一个程序和系统的特性或能力，并判断它是否达到预期的结果，软件测试就是以此为目的的任何行为 (Any activities aimed at evaluating an attribute or capability of a program or system)”。在此定义中的“设想”和“预期的结果”其实就是我们现在所说的用户需求或功能设计。他还把软件的质量定义为“符合要求”。他的核心观点是：测试方法是试图验证软件是“工作的”，所谓“工作的”就是指软件的功能是按照预先的设计执行的，采用正向思维，针对软件系统的所有功能点，逐个验证其正确性。软件测试业界把这种方法看作软件测试的第一类方法。

尽管如此，这一方法还是受到很多业界权威的质疑和挑战。代表人物是 Glenford J. Myers（代表论著《The Art of Software Testing》）。他认为测试不应该着眼于验证软件是工作的，相反应该首先认定软件是有错误的，然后用逆向思维去发现尽可能多的错误。他还从人的心理学的角度论证，如果将“验证软件是工作的”作为测试目的，非常不利于测试人员发现软件的错误。于是他于 1979 年提出了他对软件测试的定义“测试是为发现错误而执行的一个程序或者系统的过程 (The process of executing a program or system with the intent of finding errors)”。这个定义，也被业界所认可，经常被引用。这就是软件测试的第二类方法，简单地说就是验证软件是“不工作的”，或者说是错误的。Myers 认为，一个成功的测试必须是发现 Bug 的测试，不然就没有价值。这就如同一个病人（假定此人确有病），到医院做一项医疗检查，结果各项指标都正常，那说明该项医疗检查对于诊断该病人

的病情是没有价值的，是失败的。Myers 提出的“测试的目的是证伪”这一概念，推翻了过去“为表明软件正确而进行测试”的错误认识，为软件测试的发展指出了方向，软件测试的理论、方法在之后得到了长足的发展。第二类软件测试方法在业界也很流行，受到很多学术界专家的支持。

然而，对“测试的目的是证伪”这一概念的理解也不能过于片面。在很多软件工程学、软件测试方面的书籍中都提到一个概念“测试的目的是寻找错误，并且是尽最大可能找出最多的错误”。这很容易让人们认为测试人员就是“挑毛病”的，而由此带来诸多问题。大家熟悉的 Ron Patton 在《软件测试》（已有中文版）一书提出一个明确而简洁的定义“软件测试人员的目标是找到软件缺陷，尽可能早一些，并确保其得以修复”。但这样的定义具有一定的片面性，带来的结果是：

- 若测试人员以发现缺陷为唯一目标，而很少去关注系统对需求的实现，测试活动往往会存在一定的随意性和盲目性；

- 若有些软件企业接受了这样的方法，以 Bug 数量作为考核测试人员业绩的唯一指标，也不太科学。

总的来说，第一类测试可以简单抽象地描述为这样的过程：在设计规定的环境下运行软件的功能，将其结果与用户需求或设计结果相比较，如果相符则测试通过，如果不相符则视为 Bug。这一过程的终极目标是将软件的所有功能在所有设计规定的环境全部运行并通过。在软件行业中一般把第一类方法奉为主流和行业标准。第一类测试方法以需求与设计为本，因此有利于界定测试工作的范畴，更便于部署测试的侧重点，更有针对性。这一点对于大型软件的测试，尤其是在时间和人力资源有限情况下显得尤为重要。

而第二类测试方法与需求和设计没有必然的关联，更强调测试人员发挥主观能动性，用逆向思维方式，不断思考开发人员理解的误区、不良的习惯、程序代码的边界、无效数据的输入以及系统的各种弱点，试图破坏系统、摧毁系统，目标就是发现系统中各种各样的问题。这种方法往往能够发现系统中存在的更多缺陷。

到了 20 世纪 80 年代初期，软件行业进入了大发展时期，软件趋向大型化、高复杂化，软件的质量要求也越来越高。这个时候，一些软件测试的基础理论和实用技术开始形成，并且人们开始为软件开发设计了各种流程和管理方法，软件开发的方式也逐渐由混乱无序的开发过程过渡到结构化的开发过程，以结构化分析与设计、结构化评审、结构化程序设计以及结构化测试为特征。人们还将“质量”的概念融入其中，于是软件测试定义也发生了改变，它不再单纯是一个发现错误的过程，而是作为软件质量保证（SQA）的主要职能，包含软件质量评价的内容。Bill Hetzel 在《Complete Guide of Software Testing》（软件测试完全指南）一书中指出“测试是以评价一个程序或者系统属性为目标的任何一种活动。测试是对软件质量的度量”。这个定义至今仍被引用。软件开发人员和测试人员开始坐在一起探讨软件工程和测试问题。软件测试已有了行业标准（IEEE/ANSI）。1983 年 IEEE 提出的软件工艺术语中给软件测试下的定义是：“使用人工或自动的手段来运行或测定某个软件系统的过程，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”。这个定义明确指出：软件测试的目的是为了检验软件系统是否满足需求。它再也不是一个一次性的，也不只是开发后期的活动，而是与整个开发流程融合成一的。目前软件测试已成为一个专业，需要运用专门的方法和手段，需要专门人才和专家来承担。

1.2 软件测试的现状与前景

当今世界,信息技术是经济增长的重要推动力量,信息产业已成为国民经济的基础产业、支柱产业和先导产业,信息化成为各国经济和社会发展的趋势。

软件产业是信息产业的核心,是信息化建设的基础,是维护国家安全的重要保障。近年来,党中央、国务院高度重视软件产业的发展,国务院先后颁布了《关于印发鼓励软件产业和集成电路产业发展若干政策的通知》(国发[2000]18号)(简称“18号文件”)、《振兴软件产业行动纲要(2002年至2005年)》(简称“47号文件”)等文件。我国软件产业发展的政策环境不断改善,软件产业取得了快速发展,软件产业已成为我国电子信息产业中增长快、潜力大的产业之一,对促进社会经济发展发挥着重要作用。

在促进国家信息化方面,党中央、国务院也给予高度重视,中共中央办公厅17号文件《国家信息化领导小组关于我国电子政务建设指导意见》把电子政务建设作为今后一个时期我国信息化工作的重点。中华人民共和国国家发展和改革委员会发布的第55号令《国家电子政务工程建设项目管理暂行办法》,自2007年9月1日起施行。

虽然我国软件产业和信息化工作取得了较大成绩,但与发达国家相比,还有较大差距,除了规模不够大、集中度不高、缺乏大型骨干软件企业、自主创新能力不强之外,软件及信息系统质量问题也是影响我国软件产业和行业信息化健康、持续、快速发展的重要原因,主要体现在以下几个方面。

(1) 缺乏完善测试管理和测试技术体系。

目前国内的政府部门、行业与企业都认识到软件测试在信息系统质量保证工作中起着非常重要的作用,也认识到开展软件测试工作的必要性,但在实际工作中缺乏完善测试管理和测试技术体系,常暴露测试过程难控制,测试组织不合理,测试人员不合格,测试技术陈旧落后等重大问题。我国在完善测试质量体系、控制测试进度、控制测试成本、预防测试风险等方面的工作还很艰巨。

(2) 测试理论、测试技术等方面研发落后。

我国对于软件测试理论、测试技术的研究起步于“六五”期间,主要是随着软件工程的研究而逐步发展起来,但与软件发达国家相比,差距很大。在软件发达国家已经形成了测试产业,每年国际上都有各种各样的有关软件测试的年会,发表了大量的软件测试研究论文,而我国在这方面的理论研究和工程实践都比较欠缺,软件测试方面的论文发表较少。美国一些专业公司开发软件测试标准和测试工具,并占领了国际市场。因此,我们需要进行测试理论和测试关键技术的研究,自主研发具有全部知识产权的测试工具,推动我国软件测试技术水平的不断提升和人才队伍的梯队建设,促进软件产业健康、持续发展。

(3) 缺乏完善的测试环境和有效的测试手段。

很多中小型软件企业及系统集成企业,由于资金和技术问题,无法建立完善的测试环境、测试团队,以及购买测试工具,造成很多软件和系统没有经过完善的测试就投放市场,因此软件和系统就会经常出现问题,造成了用户的不满和企业维护成本的增加,甚至造成重大生产事故。而建设一个完善的测试环境和购买大量的软件测试工具,是很多中小企业短期无法承受的负担,这样就会造成恶性循环。另外,如果每一个企业都去建设这样的测

试平台,也会造成大量的闲置和重复建设。资料表明,发达国家的大型科学实验设备利用率是中国的5~6倍。目前国内使用的测试工具大都是国外生产的,价格非常昂贵,一套基本的测试管理工具、功能测试工具和性能测试工具,都需要200多万人民币投资,而操作系统、大型数据库和中间件、安全防护系统的价格更是不菲。因此,急需建立一个国家级软件和系统测试技术服务平台,为软件和系统集成企业及用户提供测试服务。

(4) 缺少必要的检验手段,“科技造假”难以杜绝。

国家每年都有大量的资金用于支持我国软件产业的科技创新以及重点软件产品与信息系统的研发,但是由于软件产品与信息系统可视性差,无法通过直观鉴别其真伪和水平,因此很多成果在验收时往往通过科研生产单位的介绍来评定,虽然很多成果通过了验收和鉴定,但是无法形成产业化,或者无法推动产业的发展,造成国家投资的极大浪费,损害了政府的形象。而且由于软件的复制成本极低,我国涉及软件知识产权的问题越来越严重,随着我国对软件知识产权保护的加强,对软件知识产权侵害的判定也越来越需要通过严格测试。

(5) 高端测试人才的培养工作刻不容缓。

软件测试人才目前是IT领域非常缺乏的人才,目前我国软件测试人才的培养存在以下盲区:一是测试人员角色定位不合理,二是缺乏专家级测试人才。

为了实现我国“十一五”软件产业发展目标,即“自主知识产权的软件比重明显提高,形成全球市场5%的产业规模和自主可持续发展能力”;为了解决当前我国软件行业发展中的瓶颈问题,进一步提升软件产品质量,提升软件产业核心竞争力,促进软件产业发展,保证信息工程建设质量和绩效;为了实现信息产业科技发展“十一五”规划(纲要)对软件、网络和信息安全技术及重大项目的规划,我们应该立足国内外先进的技术、资源和自身经验,结合我国信息产业及软件产业发展需求,致力于提升测试服务能力,促进测试理论、测试技术、测试工具的研发。这是适应我国现阶段软件产业发展需要的体现,也是为了满足行业和社会提供软件评测共性技术服务平台的需要。

1.3 软件测试工程师应具备的素质

人是测试工作中最有价值也是最重要的资源,没有一个合格的、积极的测试小组,测试就不可能实现。然而,在软件开发产业中有一种普遍的习惯,那就是让那些经验少的新手、没有效率的开发者或不适合干其他工作的人去做测试工作。这绝对是一种目光短浅的行为,对一个系统进行有效的测试所需要的技能绝对不比进行软件开发需要的少,事实上测试人员需要极其广泛的知识面与丰富的经验,他们将遇到许多开发人员不可能遇到的问题。因而,他们需要具备以下素质。

(1) 沟通能力。

一名理想的测试人员必须能够同测试涉及的所有人进行沟通,具有与技术人员(开发人员)和非技术人员(客户,管理人员)的交流能力。也就是说既要可以和用户沟通,又能同开发人员交流,不幸的是这两类人没有共同语言。与用户交流的重点必须放在系统可以正确地处理什么和不可以处理什么上。而与开发人员交流相同的信息时,就必须将这些话重新组织以另一种方式表达出来。

(2) 移情能力。

与系统开发有关的所有人员都处在一种既关心又担心的状态之中。用户担心将来使用一个不符合自己要求的系统,开发人员担心由于系统要求不正确而使他不得不重新开发整个系统,管理部门则担心这个系统突然崩溃而使其声誉受损。测试人员必须和每一类人打交道,因此需要测试人员对他们每个人都具有足够的理解和同情,只有具备了这种能力才可以将测试人员与相关人员之间的冲突和对抗减少到最低程度。

(3) 技术能力。

总体而言,开发人员对那些不懂技术的人持一种轻视态度。一旦测试小组的某个成员作出了一个错误的判定,那么他们的不可信度就会立刻被传扬出去。一个测试人员必须既能理解被测软件系统的概念又要会使用工程中的那些工具。要做到这一点需要至少几年的编程经验,这些经验有助于较深入地理解软件开发过程,并使测试人员能从开发人员的角度正确地评价自己,简化对自动测试工具的学习曲线。

(4) 自信心。

开发人员指责测试人员出了错是常有的事,测试人员必须对自己的观点有足够的自信心。如果一味容许别人对自己指东指西,就不可能完成更多的工作了。

(5) 外交能力。

当测试人员告诉开发人员软件出了错时,有必要使用一些外交方法,而且机智老练和外交手法有助于维护与开发人员的协作关系。如果采取的方法过于强硬,对测试人员来说,在以后与开发部门合作方面就相当于“赢了战争却输了战役”。

(6) 幽默感。

在遇到狡辩时,一个幽默的批评将是很有帮助的。

(7) 很强的记忆力。

一个理想的测试者应该有能力将以前曾经遇到的类似错误从记忆深处挖掘出来,这一能力在测试过程中的价值是无法衡量的。因为许多新出现的问题与已经发现的问题相差无几。

(8) 耐心。

一些质量保证工作需要难以置信的耐心。有时需要花费惊人的时间去分离、识别和分析一个错误。这个工作是那些坐不住的人无法完成的。

(9) 怀疑精神。

可以预料,开发人员会尽他们最大的努力将所有的错误解释过去。测试人员必须听取每个人的说明,但他必须保持怀疑,直到他自己看过以后。

(10) 自我督促。

干测试工作很容易使人变得懒散。只有那些具有自我督促能力的人才能够使自己每天正常地工作。

(11) 洞察力。

一个好的测试工程师应具有“测试是为了破坏”的观点,并具有善于捕获用户观点的能力,以及对质量的高要求,对细节的关注能力。

第2章 软件测试基础

本章主要阐述软件测试的定义，软件缺陷的概念，指出了软件测试的目的和原则，并从软件开发阶段、组织划分、测试技术、测试内容等不同方面介绍了不同的软件测试类型；然后结合软件开发全过程对软件测试的过程模型进行了说明，软件测试过程模型可用于测试人员和机构管理测试过程；最后，简要介绍了软件质量保证的基本概念和工作内容说明，以帮助读者了解软件质量保证和软件测试的关系。

2.1 软件测试的基本概念

软件测试是什么，是在开发快完成时对程序进行找错吗？其实不然，软件测试不仅仅只是找错而已，还需要有计划地进行，同时发现软件缺陷越早，修改的成本就越低。那么怎样才能准确又及时地发现软件缺陷呢？这首先要从软件的生命周期讲起。

2.1.1 软件生命周期

软件生命周期是从软件的产生到报废的生命周期，生命周期中有问题定义、可行性分析、总体描述、系统设计、编码、调试和测试、验收与运行、维护升级、废弃等阶段。一般来说，软件的整个生命周期可以分为如下 6 个阶段。

1. 问题的定义及规划

此阶段需要软件开发方与需求方共同讨论，主要确定软件的开发目标及其可行性。

2. 需求分析

在确定软件开发可行的情况下，对软件需要实现的各个功能进行详细分析。需求分析阶段是一个很重要的阶段。这一阶段做得好，将为整个软件开发项目的成功打下良好的基础。“唯一不变的是变化本身”，同样，需求也是在整个软件开发过程中不断变化和深入的，因此必须制订需求变更计划来应付各种变化，以保障整个项目的顺利进行。

3. 软件设计

此阶段主要根据需求分析的结果，对整个软件系统进行设计，如系统框架设计、数据库设计等。软件设计一般分为总体设计和详细设计。扎实的软件设计工作将为软件程序编写打下良好的基础。

4. 程序编码

此阶段是将软件设计的结果转换成计算机可运行的程序代码。在程序编码阶段必须制订统一并符合标准的编写规范，以保证程序的可读性、易维护性，并提高程序的运行效率。

5. 软件测试

在软件设计完成后要经过严密的测试，以发现软件在整个设计过程中存在的问题并加以纠正。整个测试过程分单元测试、组装测试、确认测试、系统测试和验收测试 5 个阶段进行。测试的方法主要有白盒测试和黑盒测试两种。在测试过程中需要建立详细的测试计划并严格按照测试计划进行测试，以减少测试的随意性。

6. 运行维护

运行维护是软件生命周期中持续时间最长的阶段。在软件开发完成并投入使用后，由

于多方面的原因,软件不能继续适应用户的要求。要延续软件的使用寿命,就必须对软件进行维护。软件的维护包括纠错性维护和改进性维护两个方面。

软件生命周期模型的发展实际上体现了软件工程理论的发展。在最早的时候,软件的生命周期处于无序、混乱的情况。这种情况直接导致了软件危机的出现,人们开始寻找产生危机的内在原因,发现其原因可归纳为两方面:一方面是由软件生产本身的复杂性导致的,另一方面是由软件开发所使用的方法和技术的多样性导致的。

为克服软件危机,人们研究和借鉴了工程学的某些原理和方法,并形成了一门新的学科,即软件工程学,但由于软件个性化的要求非常突出,在一些快速变化的商业环境下,不断变更的需求可能是一种残酷的事实,而这些都会严重影响软件产品的质量。因此,我们有必要讨论如何通过测试提高软件产品的质量,尤其是如何在早期介入测试并发现问题等。

2.1.2 软件测试的定义

软件测试是软件生命周期中的一个重要阶段,是软件质量保证的关键步骤。通俗地讲,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进行最终审查的活动。1983年IEEE在软件工程术语中提出的软件测试的定义是“使用人工或自动的手段来运行或测定某个软件系统的过程,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”。这个定义明确指出:软件测试的目的是为了检验软件系统是否满足需求。

从用户的角度来看,普遍希望通过软件测试暴露软件中隐藏的 errors 和缺陷,所以软件测试应该是“为了发现错误而执行程序的过程”。或者说,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,以发现程序错误或缺陷。

G.J.Myers 在《The Art of Software Testing》(软件测试之艺术)中指出:

- 测试是为了证明程序有错,而不是证明程序无错;
- 一个好的测试用例是在于它能发现至今未发现的错误;
- 一个成功的测试是发现了至今未发现的错误的测试。

实际上,这里暗示了“软件测试”在不同侧面上的含义,也就使人们对软件测试有不同的定义和不同的理解。

软件是由文档、数据以及程序组成的,那么软件测试就应该是对软件形成过程的文档、数据以及程序进行测试,而不仅仅是对程序进行测试。因此,软件测试的概念应进一步延伸。

从软件工程的角度出发,软件测试是软件开发过程的重要组成部分,是用来确认一个程序的品质或性能是否符合开发之前所提出的一些要求。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审查,是保证软件质量的关键步骤。软件测试在软件生命周期中横跨两个阶段:通常在编写出每一个模块之后就对它做必要的测试(称为单元测试)。编码和单元测试属于软件生命周期中的同一个阶段。在结束这个阶段后对软件系统还要进行各种综合测试,这是软件生命周期的另一个独立阶段,即测试阶段。

为提高测试的效率,在日益复杂的软件产品开发过程中,我们有时需要及时对软件开发人员开发的阶段性产品或程序片断进行测试,即针对软件开发人员每天提供的程序版本,软件测试人员需要及时响应,确保程序可以涉及的基本功能覆盖面而不出现明显的错误,即进行最基本的功能性检测,这也是目前广为关注的一种测试行为,称为冒烟测试。具体地说,冒烟测试就是在每日 build 建立后,对系统的基本功能进行简单的测试。这种测试强调功能的覆盖率,而不对功能的正确性进行验证。从这一点看它与所谓的“接受性(验收)测试(Acceptance Test)”非常相似。不同之处就在于它们执行的频率和被测的版本不同。

随着人们对软件工程化的重视以及软件规模的日益扩大,软件分析、设计的作用越来越突出,而且有资料表明,60%以上的软件错误并不是程序错误,而是分析和设计错误,因此做好软件需求和设计阶段的测试工作就非常重要。这就是目前所提倡的测试概念扩大化,提倡软件全生命周期测试的理念。

综合分析软件生命周期,可知在软件开发的全过程中都有可能产生错误,因此必须强化对整个软件开发生命周期的测试工作。正是为了更早地发现问题,所以将软件测试延伸到需求评审、设计审查活动中去,延伸后的软件测试,被认为是一种软件测试的广义概念。

如果从标准论来看软件测试,可以认为软件测试就是“验证(Verification)”和“确认(Validation)”活动构成的整体,即软件测试 = V&V。

Verification 是验证,是通过提供客观证据检验规定的要求是否得到满足,也就是说对输入与输出进行比较。Validation 是确认,是在验证好的基础上,预期的应用要求是否得到满足,也就是说,在确认时应考虑应用的条件范围要远远大于输入时确定的范围。这一般是由客户或代表客户的人执行。

Boehm 曾经很明确地指出:

Validation: Are we building the right product?

Verification: Are we building the product right?

因而可以说软件测试主要工作内容是验证(Verification)和确认(Validation),现作进一步说明。

(1) 验证(Verification)是保证软件正确地实现了一些特定功能的一系列活动,即保证软件做了所期望的事情。验证的主要内容有以下几点:

- ▣ 检验软件生命周期中的一个给定阶段的产品是否达到上一阶段所确立的需求的过程;
- ▣ 程序正确性的形式证明,即采用形式理论证明程序符号设计规约规定的过程;
- ▣ 评价、审查、测试、检查、审计等各类活动,或对某些项的处理、服务,以及文件等是否和规定的需求相一致进行判断并提出报告。

(2) 确认(Validation)是一系列的活动和过程,目的是证实在一个给定的外部环境中软件逻辑的正确性,即保证软件以正确的方式工作。确认的主要内容如下:

- ▣ 静态确认,不在计算机上执行程序,通过人工或程序分析来证明软件的正确性;
- ▣ 动态确认,通过在计算机上执行程序进行分析,测试程序的动态行为,以检验软件是否存在问题。

针对软件的验证和确认活动伴随着整个软件的生命周期,覆盖了软件需求分析、设计、编码、测试、运行等各个阶段。

(1) 需求分析阶段测试。

在需求分析阶段,需求人员会对用户的需求进行详细分析,形成产品说明书,更好的