

高等学校计算机专业规划教材

软件测试教程

第2版

宫云战 主编

赵瑞莲 张威 王雅文 等编著

Software Testing
Second Edition



机械工业出版社
China Machine Press

高等学校计算机专业规划教材

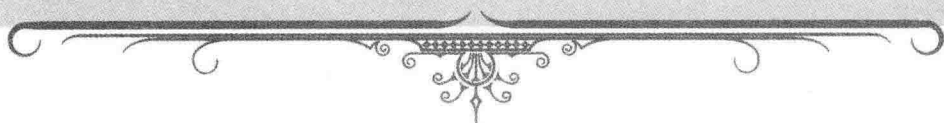


软件测试教程

第2版

宫云战 主编

赵瑞莲 张威 王雅文 等编著



*S*oftware Testing
Second Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

软件测试教程 / 宫云战主编. —2 版. —北京: 机械工业出版社, 2016.4
(高等学校计算机专业规划教材)

ISBN 978-7-111-53270-5

I. 软… II. 宫… 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 (2016) 第 057157 号

本书全面介绍软件测试的基本原理和一般方法, 同时阐述近几年出现的一些新的软件测试方法, 并结合实例介绍了目前比较流行的软件测试工具。全书共分 10 章, 分别为: 软件测试概述、软件缺陷、黑盒测试、白盒测试、基于缺陷模式的软件测试、集成测试、系统测试、主流信息应用系统测试、软件评审、测试管理。本书基本上涵盖了当今软件测试技术的全部内容, 软件测试工具是结合软件测试技术来撰写的, 并融合在每一个章节中。

本书可作为高等院校计算机专业本科、研究生的教材, 也可以作为软件测试与软件质量保障工程师的参考书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴 怡

责任校对: 殷 虹

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2016 年 6 月第 2 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 18.5

书 号: ISBN 978-7-111-53270-5

定 价: 45.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

教学建议

教学内容	学习要点及教学要求	课时安排(参考学时)
第1章 软件测试概述	- 了解软件可靠性问题 - 了解软件测试的目的和意义 - 掌握软件测试过程 - 掌握软件测试与软件开发的关系 - 了解软件测试的发展历程和现状 - 了解我国软件测试产业现状 - 了解软件测试工具	2
第2章 软件缺陷	- 掌握软件缺陷的基本概念 - 了解软件缺陷的种类和产生原因 - 了解软件缺陷数目的估计方法 - 掌握软件缺陷管理流程 - 了解常用的缺陷管理工具	2
第3章 黑盒测试	- 了解黑盒测试的基本概念 - 掌握等价类划分方法及在测试中的应用 - 掌握边界值分析法及在测试中的应用 - 掌握因果图法及在测试中的应用 - 了解决策表法及在测试中的应用 - 了解常用的黑盒测试工具	6
第4章 白盒测试	- 掌握控制流测试方法 - 掌握数据流测试方法 - 掌握代码审查方法 - 掌握代码走查方法 - 了解程序变异测试方法 - 了解常用的白盒测试工具	6
第5章 基于缺陷模式的测试	- 了解基于缺陷模式的软件测试的概念 - 了解基于缺陷模式的软件测试指标 - 掌握常见的缺陷模式,包括故障模式、安全漏洞模式、缺陷模式和规则模式 - 熟悉基于缺陷模式的软件测试系统(DTS)	6
第6章 集成测试	- 了解集成测试的概念、层次及原则 - 掌握常用的集成测试策略 - 掌握集成测试用例设计方法 - 了解基础集成测试过程 - 了解面向对象的集成测试	4

(续)

教学内容	学习要点及教学要求	课时安排(参考学时)
第7章 系统测试	• 掌握性能测试的概念及方法 • 掌握压力测试的概念及方法 • 了解容量测试的概念及方法 • 掌握健壮性测试的概念及方法 • 掌握安全性测试的概念及方法 • 掌握可靠性测试的基本概念及模型 • 了解恢复性测试与备份测试的概念 • 了解协议一致性测试的概念及方法 • 了解兼容性测试的概念 • 了解安装性测试的概念 • 掌握可用性测试的概念及方法 • 了解配置性测试的概念及方法 • 了解文档测试的概念及方法 • 了解 GUI 测试的概念及方法 • 了解回归测试的概念及方法 • 了解系统测试工具及其应用	6
第8章 主流信息应用系统测试	• 掌握 Web 应用系统测试的概念及方法 • 了解数据库测试的概念及方法 • 了解嵌入式系统测试的概念及方法 • 掌握游戏软件测试的概念及方法 • 掌握移动应用测试的概念及方法 • 了解常用的移动应用测试工具 • 了解云测试的概念及方法	6
第9章 软件评审	• 了解软件评审的目的、组织及管理 • 掌握需求评审的概念及评审细则 • 掌握概要设计评审的概念及评审细则 • 掌握详细设计评审的概念及评审细则 • 了解数据库设计评审的概念及评审细则 • 掌握测试评审的概念及评审细则	2
第10章 测试管理	• 了解测试管理体系 • 掌握测试的组织管理、过程管理、资源和配置管理及文档管理 • 掌握测试管理的原则 • 了解常用的测试管理工具	2

再版说明

随着一些新技术如物联网、云计算、大数据、移动服务等出现，使得人们要探索一些新的软件测试技术来适应。软件规模及复杂性不断增大，软件可靠性的矛盾变得越来越突出，自动化软件测试变得十分急迫。大量的软件测试工具在市场上不断涌现，目前在国内市场上销售的软件测试工具有上百种，同时在网上也有数百个开源软件测试工具。

可喜的是，国内软件测试技术发展迅速，软件测试产业已有 30 万人以上的产业队伍、3000 家以上的软件评测机构，70% 以上的中国软件企业有专业测试队伍，市场规模达数千亿，而且可以预测，在未来的 10 年都将保持 20% 以上的增长速度。

和美国相比，虽然国产软件测试工具仍有不小差距，但根据我们的初步了解，国产软件测试工具市场的占有率要远大于国产基础软件的市场占有率。目前在国产软件测试工具中，年产上千万的估计有二十余个，在软件测试所涉及的技术问题中，或多或少都有国产的工具。

本书在 2008 年第 1 版的基础上介绍一些新理论、新方法和新技术。如增加了软件缺陷概论；增加了软件代码审查和走查，这些方法虽然比较原始，但它们是发现软件缺陷很有效的办法；对目前信息技术主流应用系统（如 Web 系统、嵌入式系统、数据库系统、游戏软件和云平台）的测试也做了介绍。本书还有一个特点是突出了国产软件测试工具的介绍，在相关的章节介绍了我们自己研发的软件缺陷检测工具 DTS、软件单元测试工具 CTS，以及关键科技有限公司研发的测试管理工具 KTFlow，这三个工具在我国市场上都有一定的占有率，它们是能和美国工具相抗衡的产品。随着国产软件测试工具的不断完善，我们相信将来本书再版时 50% 以上的工具都是国产的，这也是软件测试同行奋斗的目标。

王雅文博士具体负责了再版本书的全过程，从内容组织到人员分工做了大量工作，同时她和邢颖、张威、万琳和张旭舟都参与了部分章节的撰写和修改。最后宫云战教授统稿并审阅了全书。

限于作者的水平，书中对某些问题的论述可能是肤浅的，也可能存在缺陷，恳请读者批评指正。

宫云战

2015 年 11 月 3 日于北京

第 1 版序

在过去的几十年中，软件技术得到了快速发展，软件系统的应用已经遍布社会的各个领域，成为人类改造自然不可或缺的重要组成部分。

一方面，软件的应用给社会带来了巨大的进步，大大提升了人们改造自然的能力。另一方面，由于软件的故障、漏洞等因素导致的软件不可信的程度也变得更加突出，这在很大程度上制约了软件技术的发展和软件系统的使用。软件的不可信问题正由一个纯粹的技术问题向社会问题转变，已到了非解决不可的地步。在诸如航空、航天、电信、医疗、金融等众多安全第一的领域，软件错误造成的危害是触目惊心的。

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、复杂性等因素，使得难以证明软件是正确的。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。

软件测试是发现软件缺陷，提高软件可信性的重要手段。在过去的三十年中，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较快的发展。特别是近十年来，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走向成熟，软件测试对错误与缺陷的发现能力、软件测试工具的自动化程度都得到了大幅度的提升。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兴起，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。

本书作者长期以来一直从事软件测试技术的研究和教学，对软件测试技术有比较深刻的理解，对软件测试的教学有比较好的把握。希望这本书为大家学习、理解软件测试技术提供有益的参考。



中国科学院
中国工程院 院士
北京邮电大学 教授
2008年7月

第 1 版前言

4 年前, 宫云战教授和赵瑞莲教授分别写过一本名为《软件测试》的书。同 4 年前相比, 软件测试技术与软件产业得到了快速发展, 主要表现在: 社会对其认识更加深刻、需求增大; 我国的软件评测企业大幅度增加, 目前已有近 2 000 家, 各个行业、各个省、发达地区的各个市都建立了软件评测中心; 软件测试从业人员已达 10 数万人, 我国软件测试产业产值已经达到上百亿元; 国际上的 IT 巨头, 如 IBM、微软等, 都在从事与软件测试相关的工作, 众多的 IT 企业都在中国建立了软件测试外包为主导的软件企业; 以软件测试工具和软件测试服务为核心的软件测试产业每年都在以超过 20% 速度递增; 软件测试学术活动异常活跃, 新的测试方法和测试工具不断出现。相比之下, 原来书中有些内容虽然理论性强, 但实用价值不大, 而有些内容则处于被淘汰阶段。所有这些因素都促使我们认为有必要重新撰写本书。

本书叙述软件测试的一般原理和各种基本方法, 包括基本的白盒测试、黑盒测试和集成测试方法, 并结合近几年软件测试技术的发展, 重点介绍了目前国际上一些比较流行的软件测试方法与软件测试工具, 包括:

1) 面向缺陷模式的软件测试技术: 该技术以其缺陷检测效率高、准确, 自动化程度高、易学等特点, 在过去的几年中得到迅速发展, 目前大约有 80 多个与该技术相关的工具。在美国, 以该工具为基础的软件测试服务取得了很大的成功, 成为美国一种主流软件测试技术。目前, 随着缺陷模式的不断增加, 该技术将有更广阔的应用前景。本书叙述了该技术的一般方法以及作者应用该技术开发的一款软件测试系统——缺陷测试系统(DTS)。

2) 软件评审: 软件评审发现缺陷的效率高且比较经济, 是目前常用的提高软件质量的方法, 已在许多大型软件开发中得到了印证。本书详细叙述了软件评审的内容及如何组织软件评审。

3) 随着软件开发规模的扩大、复杂程度的增加, 为了尽可能多地找出程序中的故障, 开发出高质量的软件产品, 必须对测试工作进行组织策划和有效管理, 并采取系统的方法建立起软件测试管理体系。通过它们对测试活动进行监管和控制, 以确保软件测试在软件质量保证中发挥应有的关键作用。

4) 软件测试工具是提高软件测试效率与质量的重要手段, 在过去的几年中, 在软件开发过程的各个阶段, 产生了大量的软件测试工具, 一些新技术的使用, 也使得软件测试工具的自动化程度得到了大幅度的提高。本书介绍了目前多种主流的软件测试工具。

5) 近几年来, 随着 IT 的发展, 与软件系统交互的相关技术也越来越多, 包括网络、协议、安全性、界面等, 所有这些方面都需要测试, 而这些测试和基本的软件测试是不同

的。本书全面论述了软件系统以及与此相关的系统测试。

赵瑞莲教授编写了本书的第1、2、8章，赵会群教授编写了第6章，张威教授、万琳副教授编写了第3、5、7章，杨朝红博士编写了第4章，全书由宫云战教授统稿、审查。

限于作者的水平，书中对某些问题的论述可能是肤浅的，也可能存在错误，恳请读者批评指正。

宫云战

2008年5月4日于北京

目 录

教学建议

再版说明

第1版序

第1版前言

第1章 软件测试概述1

1.1 计算机系统的软件可靠性问题1

1.2 软件测试的概念3

1.2.1 软件测试的定义3

1.2.2 测试用例4

1.2.3 软件测试的基本原则5

1.2.4 软件测试从业人员要求5

1.3 软件测试过程6

1.4 软件测试与软件开发的关系10

1.4.1 软件开发过程10

1.4.2 软件测试在软件开发中的
作用11

1.4.3 软件测试过程模型11

1.4.4 软件测试环境的搭建14

1.5 软件测试的发展历程和现状15

1.6 我国软件测试产业现状18

1.7 软件测试工具19

习题22

第2章 软件缺陷23

2.1 软件缺陷概述23

2.1.1 软件缺陷的定义23

2.1.2 软件缺陷分析24

2.1.3 软件缺陷的种类24

2.1.4 软件缺陷的产生25

2.1.5 软件缺陷数目估计26

2.1.6 软件测试效率分析29

2.2 软件缺陷管理31

2.2.1 缺陷管理的目标31

2.2.2 缺陷报告32

2.2.3 软件缺陷管理流程33

2.2.4 缺陷管理工具35

习题37

第3章 黑盒测试38

3.1 黑盒测试的基本概念38

3.2 等价类划分法39

3.2.1 等价类划分法的原理39

3.2.2 等价类划分法的测试运用41

3.3 边界值分析法45

3.3.1 边界值分析法的原理45

3.3.2 边界值分析法的测试运用46

3.4 因果图法48

3.4.1 因果图法的原理49

3.4.2 因果图法的测试运用50

3.5 决策表法52

3.5.1 决策表法的原理52

3.5.2 决策表法的测试运用53

3.6 黑盒测试方法的比较与选择56

3.7 黑盒测试工具介绍57

3.7.1 黑盒测试工具概要57

3.7.2 黑盒功能测试工具——QTP58

3.7.3 黑盒功能测试工具——
Selenium61

3.7.4 其他常用功能测试工具64

习题66

第4章 白盒测试	67	5.4 软件缺陷检测系统 (DTS)	149
4.1 控制流测试	67	5.4.1 DTS 系统结构	149
4.1.1 基本概念	67	5.4.2 DTS 缺陷模式描述	150
4.1.2 控制流覆盖准则	69	5.4.3 DTS 的测试界面	152
4.2 数据流测试	75	5.4.4 DTS 测试应用报告	152
4.2.1 基本概念	75	习题	153
4.2.2 数据流覆盖准则	76	第6章 集成测试	154
4.3 代码审查	77	6.1 集成测试概述	154
4.3.1 代码审查的意义	77	6.1.1 集成测试的概念	154
4.3.2 代码审查的内容	78	6.1.2 集成测试与系统测试的区别	155
4.3.3 代码审查的过程	80	6.1.3 集成测试与开发的关系	156
4.4 代码走查	81	6.1.4 集成测试的层次与原则	156
4.4.1 代码走查的意义	81	6.2 集成测试策略	157
4.4.2 代码走查小组的组成	81	6.2.1 非渐增式集成	157
4.4.3 代码走查的过程	82	6.2.2 渐增式集成	158
4.5 程序变异测试	82	6.2.3 三明治集成	161
4.5.1 程序强变异测试	82	6.3 集成测试用例设计	161
4.5.2 程序弱变异测试	84	6.4 集成测试过程	163
4.6 白盒测试工具	85	6.5 面向对象的集成测试	165
4.6.1 Emma	85	6.5.1 对象交互	165
4.6.2 C++test	91	6.5.2 面向对象集成测试的常用方法	166
4.6.3 JUnit	95	6.5.3 分布式对象测试	167
4.6.4 Testbed	96	习题	168
4.7 单元测试工具 CTS	98	第7章 系统测试	169
习题	103	7.1 性能测试	169
第5章 基于缺陷模式的软件测试	105	7.1.1 性能测试方法	169
5.1 基于缺陷模式的软件测试概述	105	7.1.2 性能测试执行	170
5.2 基于缺陷模式的软件测试指标分析	106	7.1.3 性能测试案例分析	171
5.3 缺陷模式	107	7.2 压力测试	174
5.3.1 缺陷模式概述	107	7.2.1 压力测试方法	175
5.3.2 故障模式	107	7.2.2 压力测试执行	176
5.3.3 安全漏洞模式	126	7.3 容量测试	176
5.3.4 缺陷模式	143	7.3.1 容量测试方法	177
5.3.5 规则模式	149	7.3.2 容量测试执行	178

7.3.3 容量测试案例分析	178
7.4 健壮性测试	180
7.4.1 健壮性测试评价	180
7.4.2 健壮性测试案例分析	181
7.5 安全性测试	181
7.5.1 安全性测试方法	182
7.5.2 安全性测试案例分析	187
7.6 可靠性测试	189
7.6.1 可靠性测试的基本概念	190
7.6.2 软件的运行剖面	193
7.6.3 可靠性测试案例分析	197
7.7 恢复性测试与备份测试	199
7.8 协议一致性测试	200
7.8.1 协议一致性测试基本概念	200
7.8.2 协议一致性测试方法	201
7.9 兼容性测试	205
7.10 安装测试	205
7.11 可用性测试	206
7.11.1 可用性测试的概念	206
7.11.2 可用性测试方法	207
7.12 配置测试	208
7.12.1 配置测试的概念	208
7.12.2 配置测试方法	208
7.13 文档测试	209
7.13.1 文档测试的概念	209
7.13.2 文档测试方法	210
7.14 GUI 测试	212
7.14.1 GUI 测试的概念及方法	212
7.14.2 GUI 测试案例分析	214
7.15 回归测试	215
7.15.1 回归测试的概念	215
7.15.2 回归测试方法	215
7.16 系统测试工具及其应用	216
7.16.1 LoadRunner	216
7.16.2 Ttworkbench	219
7.16.3 QACenter	219

7.16.4 DataFactory	220
7.16.5 JMeter	220
习题	224

第 8 章 主流信息应用系统测试

8.1 Web 应用系统测试	226
8.1.1 Web 系统基本组成	226
8.1.2 Web 应用系统测试综述	227
8.1.3 Web 应用系统测试的实施	228
8.2 数据库测试	231
8.2.1 数据库测试概述	231
8.2.2 数据库功能性测试	232
8.2.3 数据库性能测试与原因分析	233
8.2.4 数据库可靠性及安全性测试	235
8.3 嵌入式系统测试	236
8.3.1 嵌入式软件测试策略及测试 流程	236
8.3.2 嵌入式软件测试代表工具	237
8.4 游戏测试	238
8.4.1 游戏开发与测试过程	239
8.4.2 游戏测试主要内容	240
8.4.3 游戏测试的实施	241
8.5 移动应用软件测试	242
8.5.1 移动应用测试的困难	243
8.5.2 测试类型	244
8.5.3 移动应用测试工具	245
8.6 云应用软件测试	245
8.6.1 云测试基本概念	245
8.6.2 云测试方法和技术	246
8.6.3 云测试现状及挑战	247
习题	248

第 9 章 软件评审

9.1 软件评审概述	249
9.2 需求评审	250
9.3 概要设计评审	253
9.4 详细设计评审	254

9.5 数据库设计评审.....	255	10.4 测试管理实践.....	267
9.6 测试评审.....	255	10.5 常用的测试管理工具.....	268
习题.....	258	10.5.1 TestDirector 测试管理工具.....	268
第 10 章 测试管理	259	10.5.2 JIRA 介绍.....	270
10.1 建立测试管理体系.....	259	10.5.3 国外其他测试管理工具.....	272
10.2 测试管理的基本内容.....	260	10.5.4 国产测试管理工具 KTFLOW.....	274
10.2.1 测试组织管理.....	260	习题.....	276
10.2.2 测试过程管理.....	261	附录 A 测试用例样式	277
10.2.3 资源和配置管理.....	263	附录 B 测试报告样式	279
10.2.4 测试文档管理.....	263	参考文献	280
10.3 测试管理的原则.....	265	参考网站	284

第1章 软件测试概述

随着计算机技术的飞速发展，人们对计算机的需求和依赖与日俱增。随之而来的是计算机系统的规模和复杂性急剧增加，其软件开发成本以及由于软件故障而造成的经济损失也正在增加，软件质量问题已成为人们关注的焦点。因此，许多科学家在展望 21 世纪计算机科学发展方向和策略时，把软件质量放在优先于提高软件功能和性能的位置上。

软件测试是对软件需求分析、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。随着软件系统规模和复杂性的增加，进行专业化高效软件测试的要求越来越严格，软件测试职业的价值逐步得到了认可，软件测试从业人员急剧增加，软件测试评测中心如雨后春笋般成长起来。

软件测试即是从不同的角度去发现软件中的错误，或验证软件的某些属性，一个高可信的软件系统一般需要十几种乃至几十种不同的软件测试方法，在美国，软件测试的费用一般要占到整个软件费用的 50% 以上。对数千个软件项目工程的统计表明，只有 9% 的项目是成功的（在规定的时间、规定的经费和质量下完成任务），在不成功的软件项目中，一半以上与质量有关。可见软件测试是多么重要！

1.1 计算机系统的软件可靠性问题

目前上千万、上亿乃至数十亿行代码的软件比比皆是，软件规模和复杂性的急剧膨胀，使得软件故障正逐渐成为导致计算机系统失效和停机的主要因素。下面介绍几个实例。

1. 千年虫问题

千年虫问题是一个众所周知的软件故障。20 世纪 70 年代，人们所使用的计算机存储空间很小，这就迫使程序员在开发工资系统时尽量节省存储空间，一个简单的方法是在存储日期时，只存储 2 位，如 1974 存储为 74。工资系统常依赖于日期的处理，因此他们节省了大量的存储空间。他们知道在 2000 年到来时，会出现问题，比如银行在计算利息时，是用现在的日期如“2000 年 1 月 1 日”减去客户当时的存款日期（如“1974 年 1 月 1 日”），如果年利息为 3%，那么每 100 元银行应付给客户 78 元的利息。如果年份存储问题没有得到纠正，其存款年数就变为 -74 年，客户反应付给银行利息了，这显然是不合理的。但他们认为在 20 多年内程序肯定会更新或升级，而且眼前的任务比计划遥不可及的未来更加重要。为此，全世界付出了数亿美元的代价来更换或升级类似程序以解决千年虫问题，特别是金融、保险、军事、科学、商务等领域，花费大量的人力、物力对现有的各种各样程序进行检查、修改和更新。

2. 爱国者导弹防御系统

1991 年，美国爱国者导弹防御系统首次应用在海湾战争中对伊拉克飞毛腿导弹。尽管人们对该系统的赞誉不绝于耳，但是它确实在几次对抗导弹战役中出现了失误，其中一枚在沙特阿拉伯的多哈误杀了 28 名美国士兵。分析专家发现症结在于一个软件故障。一个很

小的系统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。在多哈袭击战中，这样一个小故障造成系统被拖延 100 多个小时。

3. 美国火星登陆事故

1999 年美国宇航局火星极地登陆飞船在试图登陆火星表面时突然坠毁失踪。故障评测委员会调查分析了这一故障，认定出现该故障的原因可能是由于某一数据位被更改，并认为该问题在内部测试时应该能够解决。

为什么会这样呢？简单而言，火星登陆过程计划是：飞船在火星表面降落时，着陆伞自动打开以减缓飞船的下降速度。当飞船距离火星表面 1800 米时，丢弃着陆伞，点燃登陆推进器，缓缓降落到地面。然而，美国宇航局为了节省开销，简化了关闭着陆推进器的装置，在飞船的支撑脚部安装了一个触点开关，在计算机中设置一个数据位来控制触点，以关闭飞船燃料。显然，飞船没着陆以前，推进器就应该一直处于着火工作状态。不幸的是，在许多情况下，当飞船的支撑脚迅速打开准备着陆时，机械震动也会触发触点开关，导致设置了错误的数据位，关闭了登陆推进器的燃料，使飞船加速下降 1800 米后撞向地面，撞成碎片。

结果是灾难性的，但原因很简单。事实上，飞船发射之前，经过了多个小组的测试，其中一个小组负责测试飞船支撑脚的落地打开过程，另一个小组负责测试此后的着陆过程。前一个小组没有检测触点开关数据位，那不是他们的职责；后一个小组总是在测试之前重置计算机，清除数据位。两个小组工作的都很好，但从未在一起进行过集成 / 系统测试，接口错误没有被检测出，从而导致了这一灾难性的事故。

4. Intel 奔腾处理器芯片缺陷

在 PC 的“计算器”中输入以下算式：

$$(4195835 / 3145727) \times 3145727 - 4195835$$

如果答案不为 0，就说明该计算机使用的是带有浮点除法软件缺陷的老式 Intel 奔腾处理器。

1994 年，美国弗吉利亚州 Lynchburg 学院的一位博士在用奔腾 PC 解决一个除法问题时，发现了这个问题。他将发现的问题放在互联网上，引发了一场风暴，成千上万的人发现了同样的问题，以及其他得出错误结果的情形。万幸的是，这种情况很少出现，仅在精度要求很高的数学、科学和工程计算中才会出现。

这个事件引起人们关注的原因并不是这个软件缺陷，而是 Intel 公司解决问题的态度。

- Intel 公司的测试工程师在芯片发布之前已经发现了这个问题，但管理层认为还没严重到一定要修正，甚至公开的程度。
- 当这个软件缺陷被发现时，Intel 公司通过新闻发布和公开声明试图弱化问题的严重性。
- 当压力增大时，Intel 承诺可以更换有问题的芯片，但要求用户必须证明自己受到了缺陷的影响。

结果舆论哗然。互联网上充斥着愤怒的客户要求 Intel 公司解决问题的呼声，新闻报道将 Intel 公司描绘成不诚信者。最后，Intel 公司为自己处理软件缺陷的行为道歉并拿出 4 亿多美元来支付更换芯片的费用。由此可见，这个小小软件缺陷造成的损失可能有多大。

5. Windows 2000 安全漏洞

微软曾经承认，Windows 2000 操作系统远程服务软件中存在安全漏洞，这些漏洞可能

导致3种不同的安全隐患——拒绝服务、权限滥用和信息泄漏，并发布了相应的补丁软件进行修补。拒绝服务隐患可能导致DOS受到攻击，使得合法用户无法远程登录系统；而权限滥用和信息泄漏隐患则涉及系统权限管理，有可能使攻击者控制Windows 2000系统，从而在计算机上添加用户，删除或安装组件，破坏数据或执行其他操作。

据美国军方证实，一名联机攻击者利用微软网络软件中的一个缺陷控制了其国防部的一个服务器接口。尽管美国陆军网络技术事业部称受到攻击的军事网站不属于军方，但他们强调陆军会很认真地对待这一事件。这个缺陷也使微软的安全团队大吃一惊，因为没有一名安全研究人员发现这个问题，继而造成了恶劣的影响。

2008年北京奥运会和2012年伦敦奥运会的票务系统都出现过软件故障，上海证券交易所、伦敦证券交易所等也都出现过软件故障，巨额的财产损失和多条生命的代价让人们开始重视软件质量。类似的例子可举出很多。大大小小的软件故障几乎每天甚至每时每刻都在发生，只不过有些问题不那么严重罢了。

随着信息技术的飞速发展，软件产品已应用到社会的各个领域，软件质量问题已成为人们共同关注的焦点。软件开发商为了占有市场，必须把软件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。用户为了保证自己业务的顺利完成，当然希望选用优质的软件。质量欠佳的软件产品不仅会使开发商的维护费用和用户的使用成本大幅增加，还可能产生其他的责任风险，造成公司信誉下降。在一些关键领域的应用系统，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、军事防御系统和核电站安全控制系统中，对软件质量提出了更高的要求。使用质量欠佳的软件，还可能造成灾难性的后果。毋庸置疑，如何提高软件质量，如同如何提高软件生产率一样，已成为整个软件开发过程中必须始终关心和设法解决的问题。

1.2 软件测试的概念

由于人的主观认识常常难以完全符合客观现实，与工程密切相关的各类人员之间的沟通和配合也不可能完美无缺。因此对于软件来讲，不论采用什么样的技术和方法，软件中都会存在故障。即使标准商业软件里也存在故障，只是严重程度不同而已。采用新的编程语言、先进的开发方式、完善的开发过程，可以减少故障的引入，但是我们无法完全杜绝软件中的故障。这些软件故障就需要通过测试来发现，软件中的故障密度也需要通过测试来估计。

1.2.1 软件测试的定义

软件测试是对软件需求分析、设计规格说明和编码的终审，是保证软件质量的关键步骤。但对于什么是软件测试，一直未达成共识，根据侧重点不同，主要有三种描述：

定义 1.1 1983年IEEE（国际电子电气工程师协会）提出的软件工程标准术语中给软件测试下的定义是：

“使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”。

该定义包含了两方面的含义：

- 1) 是否满足规定的需求。
- 2) 是否有差别。

如果有差别，说明设计或实现中存在故障，自然不满足规定的需求。因此，这一定义非

常明确地提出了软件测试是以检验软件是否满足需求为目标。

定义 1.2 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一批测试用例，并利用这些测试用例去执行程序，以发现软件故障的过程。该定义强调寻找故障是测试的目的。

定义 1.3 软件测试是一种软件质量保证活动，其动机是通过一些经济有效的方法，发现软件中存在的缺陷，从而保证软件质量。

上述三种观点实际上是从不同角度理解软件测试，但不论从哪种观点出发，都可以认为软件测试是在一个可控的环境中分析或执行程序的过程，其根本目的是以尽可能少的时间和人力发现并改正软件中潜在的各种故障及缺陷，提高软件的质量。

软件测试目的决定了测试方案的设计，如果我们的目的是证明程序中没有隐藏的故障存在，那就会不自觉地回避可能出现故障的地方，设计出一些不易暴露故障的测试方案，从而使程序的可靠性受到极大的影响。相反，如果测试的目标是要证明程序中有故障存在，那就会力求设计出最能暴露故障的测试方案。

软件测试是一项花费昂贵的工程，测试者希望通过软件测试来提高软件的质量或可靠性，这就意味着要发现并改正程序中的错误。所以，在进行测试时不应该假定待测软件没有故障，而应该从软件中含有故障这个假定出发去测试程序，从中发现尽可能多的软件故障。因此，“一个成功的测试是发现了至今未被发现的故障的测试”，“一个好的测试用例在于发现至今尚未被发现的故障”。

1.2.2 测试用例

测试用例 (Test Case) 是为某个特殊目标而编制的一组数据，包括测试输入、执行条件以及预期结果，以便测试某个程序路径或核实是否满足某个特定需求。

测试用例目前没有经典的定义。比较通常的说法是：指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤以及预期结果等，并形成文档。

要使最终用户对软件感到满意，最有力的举措就是对最终用户的期望加以明确阐述，以便对这些期望进行核实并确认其有效性。而测试用例就是用于反映要核实的需求，这些需求可以通过不同的方式并由不同的测试员来实施。但有些情况下可能无法（或不负责）核实所有的需求，那么是否能为测试挑选最适合或最关键的需求则关系到测评项目的成败。选中要核实的需求将是对成本、风险和必要性三者权衡考虑的结果。

如何以最少的人力、资源投入，在最短的时间内完成测试，并发现软件系统的缺陷，以保证软件的质量，是软件测评机构和测试人员探索和追求的目标。每个软件产品或软件开发项目都需要有一套优秀的测试方案和测试方法。影响软件测试的因素很多，例如软件本身的复杂程度、开发人员（包括分析、设计、编程和测试的人员）的素质、测试方法和技术的运用等等。因为有些因素是客观存在的，无法避免。有些因素则是波动的、不稳定的，例如测评队伍是流动的，有经验的走了，新人不断补充进来；一个测试人员的工作也受情绪的影响等。有了测试用例，则可以保障软件测试质量的稳定。无论是谁来测试，参照测试用例实施，都能保障测试的质量，这样就可以把人为因素的影响减少到最小。即便最初的测试用例考虑不周全，随着测试的不断进行和软件版本更新，也将日趋完善。因此测试用例的设计和编制是软件测试过程中最重要的活动。测试用例是测试工作的指导，是软件测试必须遵守的