高等学校计算机应用规划教材

软件测试技术

- 软件测试的过程与分类
- 软件测试过程模型
- 测试用例的编写规范
- 白盒与黑盒测试技术
- 软件测试的主要执行阶段
- 典型的功能测试与非功能测试
- 软件缺陷报告与测试评估
- 软件测试项目的管理方法
- 软件自动化测试技术

杨怀洲　编著

清华大学出版社

高等学校计算机应用规划教材

软件测试技术

杨怀洲　编著

清华大学出版社
北　京

内 容 简 介

本书系统地介绍软件测试的基本原理与方法，重点讲解软件测试的基本技术、测试用例的设计方法、软件测试的主要过程、软件缺陷的报告以及测试的评估方法。同时，结合软件测试工程实践，讲解测试项目管理、自动化测试原理以及测试工具的分类和选择。书后附录部分给出了常用软件中测试术语的中英文对照、与测试相关的软件工程国家标准目录、实用的软件测试计划模板和验收测试报告模板，供读者学习参考。

本书融入作者十余年软件工程领域实践与教学经验，内容精炼实用、条理清晰并且通俗易懂。通过丰富的实例和实践要点描述，方便读者理解测试理论和技术的具体应用方法，力求使软件测试初学者可以在短时间内掌握软件测试技术核心内容，为进一步适应高级软件测试工作打下坚实基础。

本书可作为软件工程、计算机科学与技术以及相关专业的本科生教材和硕士研究生参考教材，也可以作为各类软件工程技术相关人员的参考书。

本书对应的课件可以到 http://www.tupwk.com.cn/downpage 网站下载，也可通过扫描前言中的二维码下载。

图书在版编目(CIP)数据

软件测试技术 / 杨怀洲 编著. —北京：清华大学出版社，2019
(高等学校计算机应用规划教材)
ISBN 978-7-302-52501-1

Ⅰ. ①软… Ⅱ. ①杨… Ⅲ. ①软件—测试—高等学校—教材 Ⅳ. ①TP311.55

中国版本图书馆 CIP 数据核字(2019)第 043098 号

责任编辑：胡辰浩　李维杰
装帧设计：孔祥峰
责任校对：牛艳敏
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市少明印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**17.75　　**字　　数：**455 千字
版　　次：2019 年 4 月第 1 版　　**印　　次：**2019 年 4 月第 1 次印刷
定　　价：59.00 元

产品编号：083108-01

前　言

现阶段，软件测试基础人才不足，已成为制约我国软件产业发展的瓶颈。在国内，虽然软件测试仍然处于起步阶段，但是毫无疑问，就 IT 产业发展前景来看，软件测试是软件行业中的朝阳产业。信息产业目前已成为我国的支柱性产业，特别是伴随着“互联网+”战略上升至国家战略，软件行业正在以前所未有的速度蓬勃发展，因此也极大地带动了软件测试行业的快速发展。软件测试对于软件质量保障的重要性越来越多地得到软件企业和软件研发团队的重视，专业的软件测试人才需求不断扩大，各种软件测试培训机构和网站数量不断增多，软件测试已成为 IT 产业中的一个重要行业分支。

但是与软件测试发展和人才需求不相适应的是，很多软件企业认为，大量软件测试岗位应聘者缺乏对于软件测试技术的系统培训，未能系统化地掌握软件测试正规流程，一些应聘者虽然有一些软件研发经验，但是不了解软件测试岗位需求。从软件测试人员的现状来看，也存在着很多问题。测试人员的专业知识不够扎实，只懂得一些表面上的测试技术，不能全面胜任软件测试工作，更无法胜任软件测试项目管理工作；测试人员没有建立相对完整的测试体系概念，对软件测试的基本定义和目的不清晰，不了解如何具体开展软件测试工作；忽视软件测试理论知识，认为理论知识没有用而不去深入理解软件测试的基本原理。软件测试人才知识能力结构不健全的根本原因是人才培养途径不健全，因此，急须加强高等院校软件测试技术相关课程的建设。

软件测试远比人们直观想象的复杂，测试工作具有很高的组织管理和技术难度，测试理论也比较庞杂，具有理论和实践高度联系的特点。软件测试工具相比于软件开发工具来讲，分类更为细致并且数量众多，一个软件项目的测试工作往往需要多种软件测试工具的配合使用才能达到全面和深入测试的效果。同时，高等院校软件测试技术课程的讲授或多或少地受到讲授形式和有限课时的限制，与培训机构动辄 4~6 个月的软件测试培训周期和以实验练习为主的培训方式有很大的不同。

上述实际情况使得高等院校软件测试技术课程的讲授具有一定的挑战性，需要精心编排和组织授课内容，设定合理的授课目标，力争在有限的学时内，使学生掌握软件测试的基本原理、方法和技术，熟悉软件测试的正规流程以及相关标准和规范，了解软件测试项目的管理方法，并且能够学习掌握自动化测试的原理和基本的软件测试工具，为今后更为深入地学习软件测试知识技能和胜任高级软件测试工作打下坚实基础。为此，本书在内容上进行了精心组织，摒弃了一些复杂深奥和实用性不强的理论内容，加强了对软件测试基本技术的讲解，力求通过丰富的典型实例和通俗易懂的语言，使读者快速理解重点内容、切实掌握相关难点。同时，重视理论与实践相结合，根据当前软件测试行业技术应用现状和未来发展趋势，使读者既能够系统地

掌握软件测试的基本理论和方法，又能够明晰这些理论和方法是如何在实际应用中发挥作用的。本书主要包括以下内容：

(1) 测试基础知识。在第 1 章中通过分析软件测试工程师的职业发展前景和当前我国软件测试行业现状，使读者首先了解学习本课程的意义，增强学习兴趣；介绍软件测试的发展历程、基本概念、原则和术语；详细说明软件测试的目的、分类、流程和基本的软件测试过程；细致讲解常见的软件测试模型；阐述什么是测试用例、如何正规书写测试用例、如何保障测试用例的设计质量。

(2) 测试基本技术。在第 2 章和第 3 章中结合经典实例，重点讲解常用的白盒测试技术和黑盒测试技术以及相应的测试用例设计方法，对难以掌握和应用中易错的知识点进行实例化说明。总结和分析白盒测试和黑盒测试的优缺点，在此基础上给出白盒测试和黑盒测试技术的应用策略。

(3) 测试过程。在第 4 章中，从单元测试、集成测试、系统测试和验收测试 4 个阶段详细介绍软件测试执行过程，说明各测试阶段依据的主要技术文档、参与人员、典型测试数据和采用的主要技术，对回归测试的方法和注意事项进行介绍。

(4) 功能与非功能测试。在第 5 章中，对各种典型的功能和非功能测试技术进行说明，重点讲解性能测试的分类以及常用的性能测试指标。

(5) 缺陷报告与测试评估。在第 6 章中，详细说明报告软件缺陷的方法，重点说明如何完成定量化测试评估，介绍测试总结报告的编写方法。

(6) 测试管理。在第 7 章中，介绍测试管理中一些最为重要的管理内容和相关知识，主要包括软件质量管理标准和管理体系、如何制定测试计划、测试项目中的测试文档以及测试配置管理等内容。

(7) 软件测试自动化。在第 8 章中，介绍自动化测试的原理，说明测试工具的分类和选择方法，给出一些常用测试工具的说明。

本书在编写过程中参考了很多专著、教材、论文和大量的网上资料，由于篇幅所限，一些细节之处未能一一列出。在此，向所有作者表示衷心的感谢和诚挚的敬意。由于作者专业水平有限，书中难免有缺点和欠妥之处，恳请读者批评指正，以便于今后不断修正和改进。我们的电话是 010-62796045，信箱是 huchenhao@263.net。

本书对应的课件可以到 http://www.tupwk.com.cn/downpage 网站下载，也可通过扫描下方的二维码下载。

作　者

2018 年 12 月

目　　录

第 1 章

软件测试概述

软件测试是保障软件质量的关键手段。当今社会是信息化社会，各种信息化技术高速发展，表现为软件几乎无所不在。软件不仅存在于我们的计算机中，而且几乎存在于我们日常接触和使用的所有电子设备之中。随着软件数量、规模和复杂度的增加，其质量优劣深刻影响着各行各业的发展和人们的日常生活。对软件测试技术的学习与应用越来越得到 IT 从业人员的重视。

本章介绍软件测试的行业需求、现状和发展历程，给出软件测试的基本概念、目的、分类和原则，说明软件测试过程、常见测试模型和测试用例。目的在于使读者在深入学习具体测试技术之前，首先建立起正确和全面的测试思想，理解和掌握软件缺陷、软件测试、测试过程模型和测试用例编写规范等基本软件测试知识。

1.1 软件测试行业需求与现状

近年来，我国软件测试行业一直呈现出迅猛发展的势头，留意一下 51Job 和中华英才网有关软件测试工程师的大量招聘信息即可体会一二。据国家权威部门统计，中国软件测试人才缺口高达 30 多万，并且仍以每年 20%的速度增加。这一现象背后的主要原因是，随着软件行业竞争的加剧和用户对于软件产品质量意识的逐步提升，国内软件企业都在加大对于软件质量管理的投入。我国软件行业起步较晚，大量软件企业在发展初期往往重开发而轻测试，软件质量管理意识薄弱，为抢占市场和降低运营成本，片面追求软件开发的短平快，不少企业因软件产品质量问题而导致生存极其困难。

我国软件测试行业虽然经历了近几年的快速发展，但是仍然非常薄弱，比较明显的反映是软件测试和开发工程师在人员比例上的严重失衡。国际上公认的软件测试和开发工程师人员配置比例标准是 1∶1，国外一些开发大型、复杂软件系统的成熟企业(如微软公司)，软件测试人员和开发人员的比例约为2∶1，表1-1 中是微软公司两个大型软件产品在开发过程中开发人员和测试人员的比例。

表 1-1　微软公司 Exchange 2000 和 Windows 2000 项目的测试人员和开发人员比例

	Exchange 2000	Windows 2000
项目经理	25 人	250 人
开发人员	140 人	1700 人
测试人员	350 人	3200 人

测试人员与开发人员比例	2.5∶1	1.9∶1

但是如图 1-1 所示，能够达到上述合理比例的国内企业数量仅占 7%，1∶7 以上配置比例的企业数量仍然高达 18%，这从一个侧面反映了我国软件测试行业的成熟度与国外相比还有很大的差距，同时也说明其增长空间是巨大的。随着移动计算、互联网+、物联网和大数据的兴起，我国软件产业在今后必将蓬勃发展，但是软件测试人才的极度短缺会成为制约软件产业发展的瓶颈，这一状况的改善需要一定的时间。除此之外，国内软件测试行业在对测试的理解和认识、对测试过程的正规化管理、对自动化测试工具的使用和对测试人员的培养方面也存在着诸多不足之处。

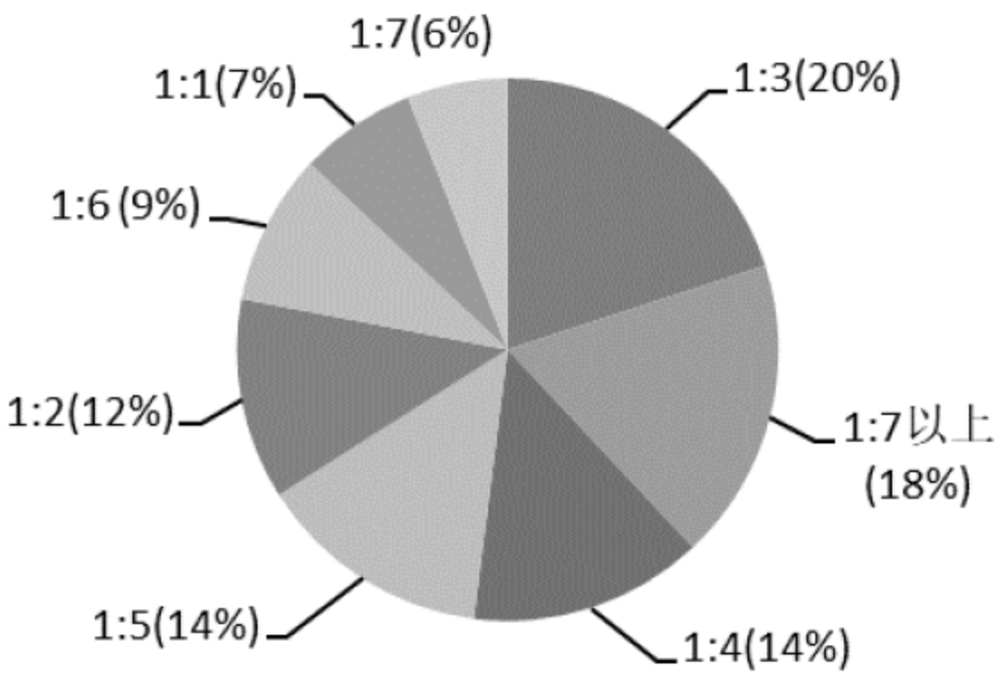


图 1-1　软件测试人员和开发人员比例

软件测试职业具有良好的发展空间和独特的职业优势，除了上述谈到的软件测试人才需求量越来越大的特点外，其职业优势还体现在以下几个方面：

(1) 职业门槛不高。通过短期系统化地学习软件测试理论和技术，就能胜任基本的软件测试工作。但需要注意的是，软件测试工作对于专业综合素质要求比较高。

(2) 职业生命周期长。软件测试工作没有年龄限制，更多的是要求经验和耐心，随着测试经验的不断丰富与积累，职业价值会越来越高。

(3) 无性别偏好。软件开发岗位因种种原因男性从业人员较多，软件测试工作对于耐心细致、条理性、沟通与交流能力等相对偏重，工作压力和强度相对较小。据统计，目前软件测试领域女性比例比男性稍多一点，维持在男女比例较均衡的状态。

(4) 多元化发展，职业空间广阔。测试人员不但需要对软件的质量进行检测，还能接触到与软件相关的各行各业，项目管理、沟通协调、市场需求分析等能力都能得到很好的锻炼，从而为自己的多元化发展奠定基础，经过一两年实践后，很容易晋升到主管、项目经理等高级职位。

如图 1-2 所示，据统计，软件测试人员所属行业主要是通信及互联网行业、应用软件行业和金融行业，所占比例分别为 41%、20%和 14%，三者总体占比达到 75%。这一数据从一个侧面反映了在我国信息化建设过程中，软件产品开发和应用色彩浓厚的企业受到社会和个人的青睐，测试人才需求量不断上升。可以预见的是，随着我国建设数字化强国战略的实施，电子、教育、政府和工业控制等行业对于软件测试人才的需求量还将不断扩大。

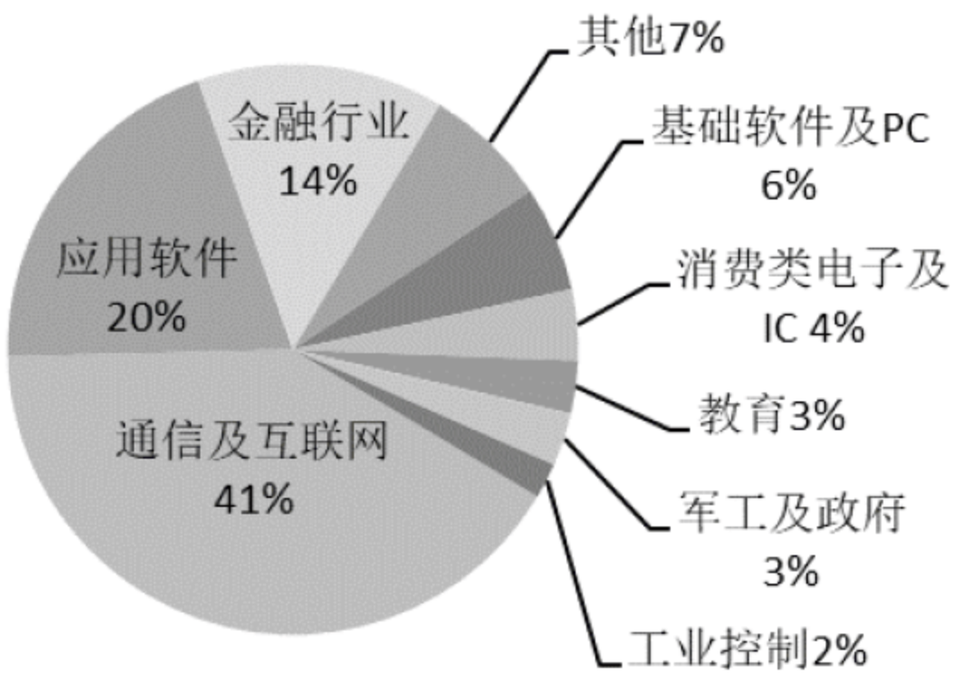


图 1-2　软件测试人员所属行业分布

从软件测试职业发展方向来看，大体可以分为技术和管理两个方向。根据测试能力和经验的不同，技术方向又可分为初级、中级和高级 3 个技术阶段。

1) 初级技术阶段

大体涵盖初级和中级软件测试工程师职位，一般是进入软件测试行业 3 年以内的常规测试从业者所经历的阶段。处于这一阶段的测试人员的工作内容主要是接受测试主管分配的任务，根据已有测试计划、流程和方案编写和执行测试用例、提交软件缺陷报告、完成阶段性测试报告以及参与部分阶段性评审工作。

初级测试工程师一般需要熟悉软件测试理论和技术、测试用例编写方法、测试流程和规范、常用测试管理工具，能够独立设计测试方案和编写测试报告，主要完成软件功能测试任务。中级测试工程师一般需要能够制定测试计划、审核功能需求和设计文档、编写自动化测试脚本，并且能够合理运用测试工具完成自动化测试，提交质量分析报告。

2) 中级技术阶段

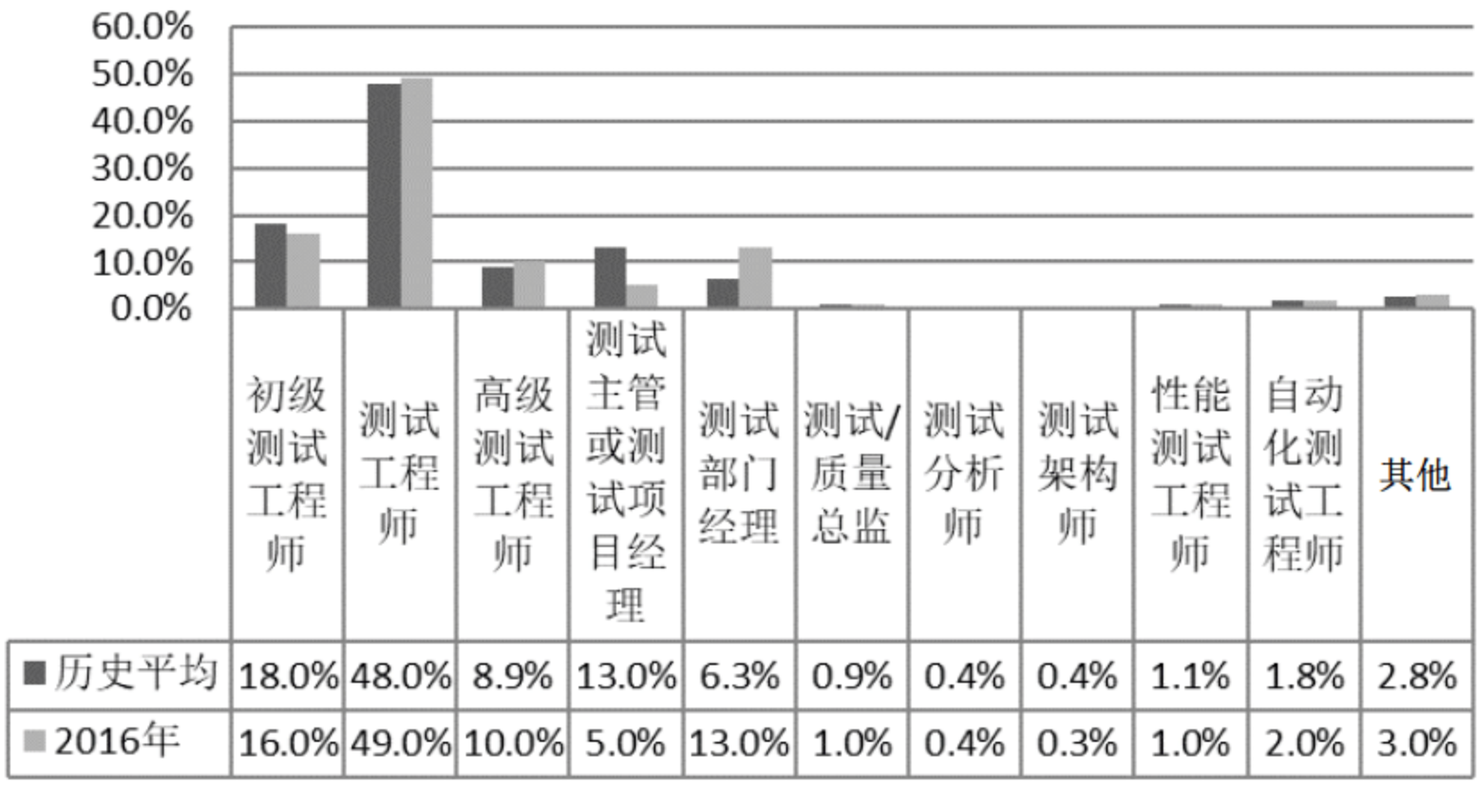
处于此阶段的软件测试从业人员已经能够胜任自动化测试工程师、白盒测试工程师、性能测试工程师职位。自动化测试工程师能够熟练运用测试工具进行软件黑盒测试。白盒测试工程师能够完成单元测试阶段的代码级测试，包括代码走读、逻辑测试、代码效率检查和覆盖率分析等。由于需要熟练掌握程序语言，因此技术要求较高。性能测试工程师需要能够在功能测试后对系统性能指标进行测试分析，综合技术能力要求非常高，要求熟练掌握软件开发、操作系统、数据库、应用服务器、网络协议等理论和技术，这样才能够准确捕捉和定位性能问题。

3) 高级技术阶段

相比于中级技术阶段的技术能力要求，高级技术阶段的自动化测试工程师不仅需要能够完成自动化测试脚本的设计和开发，还需要能够设计数据驱动，开发测试框架以及自主开发测试特定业务所需要的一些企业内部小型测试工具。对于白盒测试来讲，需要结合不同软件架构和多种开发技术，寻找最为有效的代码测试方法，并且具有对代码进行优化的能力。对于性能测试来讲，需要具备设计整体性能测试方案的能力，这就要求对主流的软件开发模式和应用系统具备丰富的知识和经验。

从技术能力上来讲，安全测试工程师已经属于高级技术阶段的职位。这是由于安全测试工

程师必须具有极高的专业理论和技术水平，对安全标准和体系、操作系统与网络、软件开发模式和架构、软件系统功能和性能测试等都要有很丰富的经验，这样才能预见软件安全漏洞并且进行尝试性攻击，完成安全性测试任务。



	初级测试工程师	测试工程师	高级测试工程师	测试主管或测试项目经理	测试部门经理	测试/质量总监	测试分析师	测试架构师	性能测试工程师	自动化测试工程师	其他
历史平均	18.0%	48.0%	8.9%	13.0%	6.3%	0.9%	0.4%	0.4%	1.1%	1.8%	2.8%
2016年	16.0%	49.0%	10.0%	5.0%	13.0%	1.0%	0.4%	0.3%	1.0%	2.0%	3.0%

图 1-3　软件测试人员职位分布

软件测试职业向管理方向发展，一般会经历测试主管、测试经理和测试总监 3 个职位。

1) 测试主管

一般由具备 3~5 年软件测试经验的人员担任，其工作职责通常是根据项目经理或测试经理的计划安排，负责调配 1~3 名测试工程师完成模块级或项目级测试工作。测试主管需要熟练掌握各种测试方法，具备过硬的测试技术，其具体工作内容往往是负责测试流程的具体实施，在思考如何对软件进行深入、全面测试的基础上完成测试设计工作。

2) 测试经理

一般由管理和技术能力都比较成熟的资深测试人员担任，负责调配业务测试、功能测试、性能测试等不同类型测试工程师，完成企业级或大型项目级软件总体测试工作的策划和实施。除此之外，还需要研究不同软件架构和开发技术下的测试方法，为测试团队成员提供业务指导。

3) 测试总监

在大型软件企业或专门提供测试服务的企业中一般会设立该职位。测试总监负责企业级全部测试及产品质量保障工作，全面管理企业的相关测试资源。

由图 1-3 的统计结果可知，虽然软件测试职位具有很多分类，但是初级和普通的测试工程师两者占比之和高达 55%，高精专的技术和管理测试人才非常稀缺，反映了 IT 行业从来都是“千军易得，一将难求”的行业特征。

1.2 软件中的 Bug

1.2.1 Bug 与软件缺陷

我们常说，软件测试就是寻找软件中的 Bug。Bug 是“虫子”的意思，现在人们将计算机

系统或程序中隐藏的问题、缺陷或漏洞统称为 Bug。在软件测试领域，我们一般使用更为正规的名词：软件缺陷(Software Defect)。

软件缺陷的含义比较宽泛，国内软件可靠性工程领域广泛使用的定义为：软件缺陷就是存在于软件(程序、数据和文档)中的那些不希望或不可接受的偏差，会导致软件产生质量问题。在这里需要注意两点：一是这里所说的偏差是指软件与用户需求的偏差；二是不仅是程序，各类软件文档中的错误也属于软件缺陷，都是 Bug！

IEEE 国际标准 729—1983 中给出的软件缺陷标准定义是：从软件产品内部看，软件缺陷是软件开发或维护过程中存在的错误、毛病等各种问题；从软件产品外部看，软件缺陷是系统所需要实现的某种功能的失效或违背。

通常认为，只要符合下面 5 条规则中的任意一条，就是软件缺陷：

(1) 软件未达到软件规格说明书中规定的功能；

(2) 软件超出软件规格说明书中指明的范围；

(3) 软件未达到软件规格说明书中指出的应达到的目标；

(4) 软件运行出现错误；

(5) 软件难以理解，不易使用，运行速度慢，或者最终用户认为使用效果不好。

在一些外文书籍中，只是将软件测试执行过程中发现的问题称为 Bug，而将软件需求和设计阶段引入的错误称为 Defect(缺陷)，将编码错误称为 Error(错误)，将软件交付用户使用过程中的错误称为 Failure(故障)。这种狭义上的概念划分我们了解即可。

Bug 一词的产生来源于一次有趣的历史事件。

1945 年 9 月 9 日，美国海军的 Grace Hopper 中尉作为程序员正在研制一台被称为 MARK Ⅱ 的计算机。那时的计算机还不是完全的电子计算机，需要使用大量的继电器完成工作。因为正值炎热的夏天，机房又位于一栋没有空调的老建筑内，因此为了散热，所有窗户都敞开着。突然，MARK Ⅱ死机了。Grace Hopper 经过排查，在计算机的继电器里，找到了一只被继电器打死的小飞蛾，这只小虫子卡住了计算机的运行。Grace Hopper 顺手将飞蛾夹在工作笔记里，并诙谐地把程序故障称为“Bug”。这一称呼后来演变成表达缺陷漏洞的计算机专业术语，人们习惯地把排除程序故障叫作“Debug”(除虫)。这一事件记录和那只飞蛾可以被称作第一个 Bug 记录手稿(如图 1-4 所示)，现在陈列在美国历史博物馆中。这就是我们今天所说的 Bug 的由来，它最初竟然真的就是“一只虫子”。

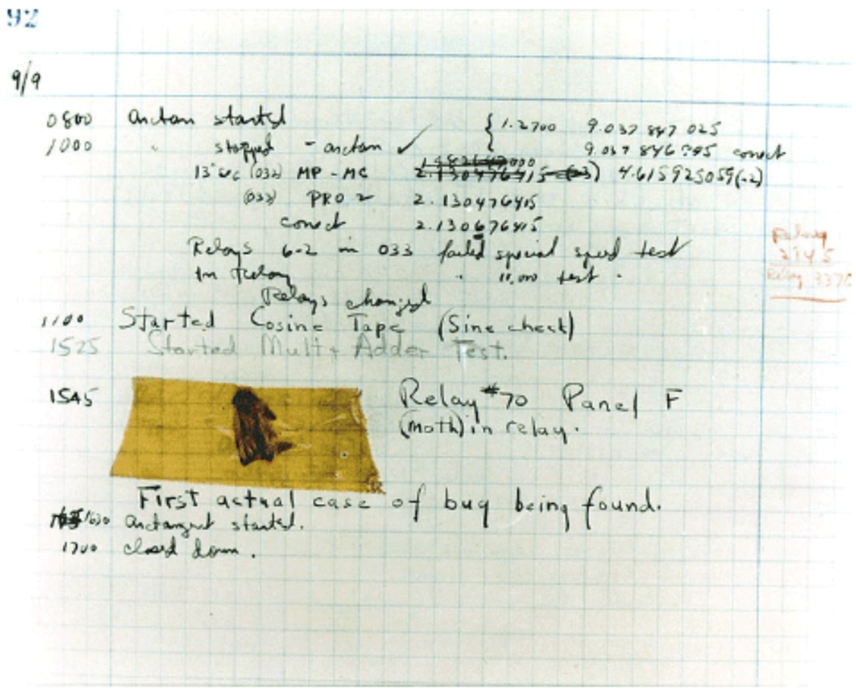


图 1-4　第一个 Bug 记录手稿

值得一提的是，Grace Hopper(如图 1-5 所示)后来成为美国海军少将，她是第一个程序语言编译器的开发者，第一个使用词语的计算机语言开发者，第一个商用编程语言 COBOL 的开发者，是与阿兰·图灵、史蒂夫·乔布斯、比尔·盖茨等一同入选“IT 界十大最有远见的人才”的唯一一位女性。2016 年，Grace Hopper 被奥巴马追授总统自由勋章，这是美国平民所能获得的最高荣誉。

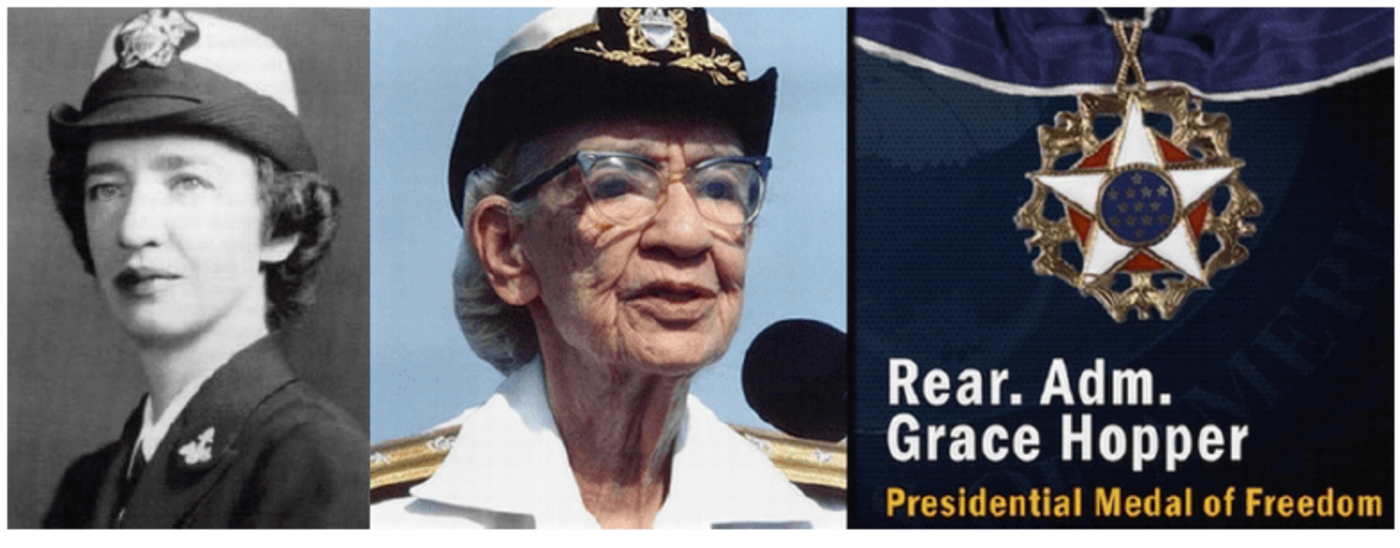


图 1-5　Grace Hopper

1.2.2　软件 Bug 的普遍性与危害性

今天，在我们的日常生活中所能碰到的软件 Bug，多到了令人难以置信的程度。就拿我们经常使用的 Windows 系统来说，它的每一个版本中都含有太多的 Bug，从微软公司每发布新的 Windows 版本后不断推出补丁程序这一现象我们就能窥知一二。据报道，早期的 Windows 95 中含有 5000 多个 Bug！即使目前最新版本的 Windows 系统也仍然有似乎无数的 Bug 被不断地发现。再看看我们手机中的 Android 系统，越用越慢，无缘无故死机，各种 Bug 频出。

据统计，每年因软件问题会让美国经济损失近 600 亿美元。软件 Bug 的普遍存在影响的不仅仅是我们的日常生活，历史上很多灾难性事件的发生都是由软件 Bug 引起的。

1979 年 11 月 28 日，新西兰航空 901 号班机因计算机控制的自动飞行系统发生故障在南极洲埃里伯斯火山撞山坠毁，机上 237 名乘客及 20 名机组成员全部罹难。

1982 年夏天，苏联西伯利亚天然气管线发生一次特大爆炸，这次爆炸对苏联经济的打击异常沉重。由于天然气管线设计十分复杂，需要一种高级自动控制软件进行控制，当时苏联人还没能掌握相关技术。苏联政府向美国公司求购遭到拒绝，随后派遣间谍进入一家加拿大公司企图盗取这一技术。美国中情局特工决定让苏联人得到一种“特殊版本”的自动控制软件，并且将这种软件提供给所有可能已被苏联间谍渗透的加拿大公司。苏联间谍果然偷走有问题的软件，导致前面提到的大爆炸。

1987 年 10 月 19 日，道琼斯指数一天之内下跌达 22.6%，创下历史单日最大跌幅，引发金融市场恐慌，股市一天就损失 5000 亿美元。很多人认为这场股灾是因程序交易引起的，计算机程序看到股价下挫，便按早就设定好的机制大量抛售股票，造成系统崩溃，导致大多数的投资者盲目跟从，从而形成恶性循环，令股价加速下跌。

1991 年 2 月 25 号，海湾战争期间，一枚美国的爱国者导弹因为基于内部时钟的时间计算缺陷，无法在沙特阿拉伯的达兰成功拦截伊拉克发射过来的一枚飞毛腿导弹。该飞毛腿导弹击中该地区的一个美军军营并导致 28 名士兵阵亡。美国审计总署提供的报告描述了该拦截失败的原因，其标题为：“爱国者导弹防御：软件缺陷导致防御系统在沙特阿拉伯达兰的拦截失败”。

1996 年 6 月 4 日，欧洲航天局发射的一架未载人的阿丽亚娜 5 号火箭，在发射升空 40 秒之后发生了爆炸。这项造价 70 亿美元的火箭项目是 10 年研发后的首次发射。损毁的火箭和货仓就价值 5 亿美元。事故调查委员会两周后给出了爆炸的起因，声称是惯性参考系统的一个软件错误引发的。

2016 年 10 月 19 日，欧洲太空总署火星登陆器“斯基亚帕雷利”坠毁在火星表面。欧洲航天局事后宣布，事故原因调查发现，由于用以测量登陆器旋转速度的数据错误，登陆系统软件认为登陆器已经着陆，提早释放了降落伞，而减速用的推进器只点火几秒就终止，当时它仍然位于火星表面上方 3.7 公里处。错误虽只持续了一秒，但足以破坏登陆器的导航系统。

以上我们列举了一些历史上发生在航空、航天、能源、金融和军事领域的典型软件缺陷案例，但这仅仅是“软件 Bug 灾难史”的一小部分！今天，随着软件无所不在，软件 Bug 也总是如影随形。软件测试就是为了发现和清除软件缺陷，生产出满足人们需要的高质量软件产品。

1.2.3　软件缺陷产生的原因

软件中有如此多的缺陷，其产生原因是多方面的。我们具体分为软件、技术、团队和管理四个方面并进行分析(如图 1-6 所示)。

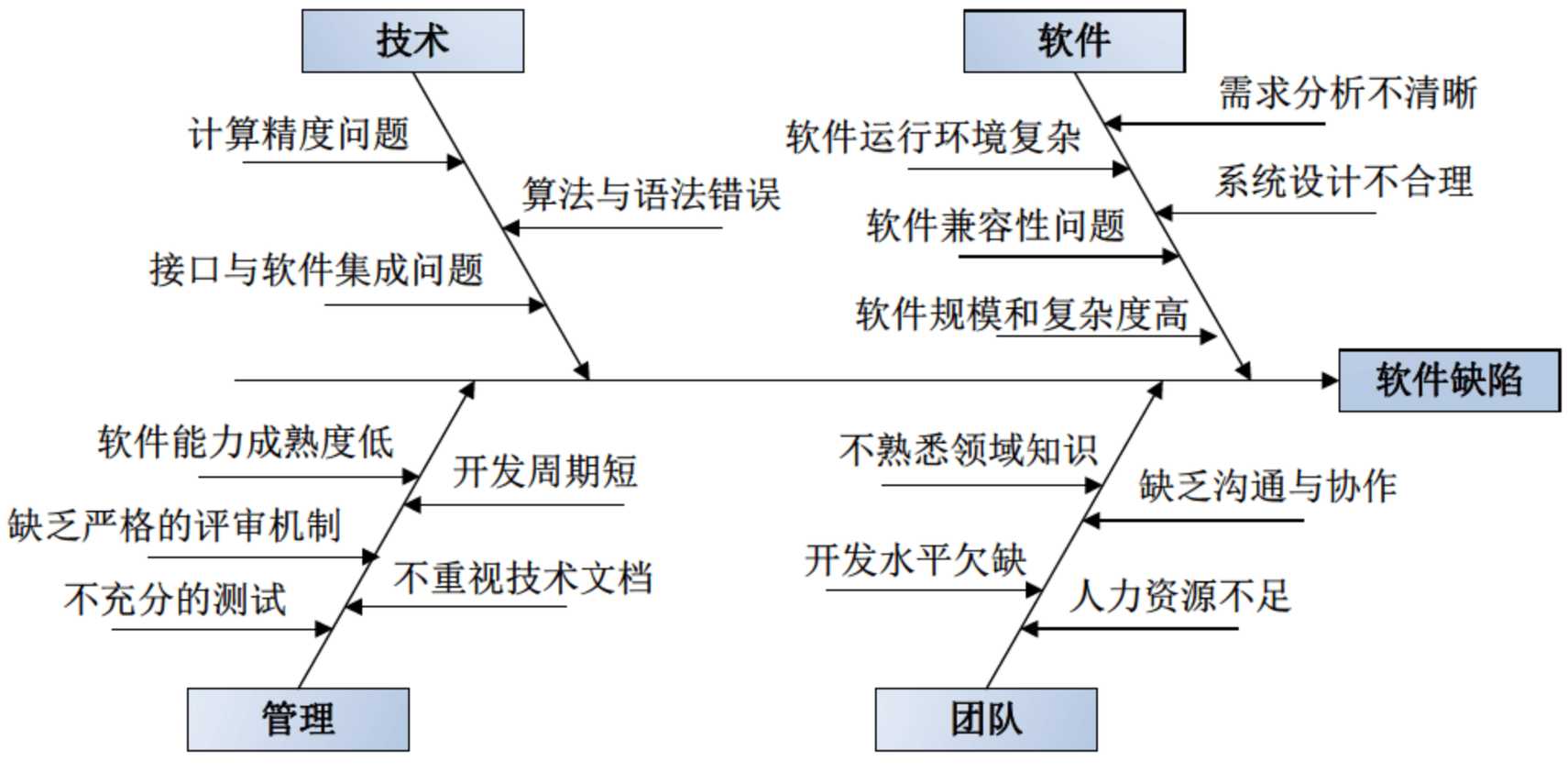


图 1-6　软件缺陷产生的原因

1) 软件自身因素

软件是具有复杂运算逻辑的产品，软件的规模越来越庞大，复杂度也越来越高，软件是由人开发的，因此不可能完美无缺，软件缺陷是难以避免的。实践表明，软件需求和设计问题是导致软件缺陷的主要原因，包括软件需求说明书编写不全面、不完整、不准确，需求分析时与用户交流不足，需求的频繁变更，以及开发人员不能很好地理解需求说明书和沟通不足等。当需求分析人员与用户沟通的时候，没有详细了解到用户的具体需求，导致需求分析不够全面。分析人员可能会误解用户需求，或者做软件分析说明书时出现误差。软件设计人员在体系结构或构件设计方面可能设计得不够合理。编程人员按照软件设计说明实现软件时，也可能在理解和沟通上出现问题，做出的产品与设计不符。除此之外，软件运行环境的复杂性以及兼容性等问题，也是软件缺陷产生的重要原因。

2) 软件涉及的具体技术问题

软件开发会涉及大量具体的技术问题，如算法、语法、软件接口、计算精度、软件安全、性能保障以及模块封装与集成，等等。对于任何一项技术，如果有处理不当之处，都会产生软件缺陷。

3) 开发团队问题

开发团队如果出现业务领域知识不足、开发水平达不到要求、人员之间缺乏沟通与协作以及软件项目团队人力资源不足的情况，都会埋下产生软件缺陷的隐患。

4) 项目管理问题

大量的软件项目难以按照计划的进度和预算完成，超预算和超进度的情况比比皆是。软件项目进度和预算的压力必然会对软件项目管理造成影响，使得原有的软件质量控制目标难以达成。保证软件质量的测试工作是相当耗时的，一个软件项目往往需要经过很多次的测试才能将软件缺陷的数量降低到可以接受的程度。此外，软件能力成熟度低、技术文档缺失、评审不严格等管理问题必然会引发大量的软件缺陷。

1.3 什么是软件测试

我们都知道测试的英文是 Test，源于拉丁语 Testum，原意是罗马人使用的一种陶罐，用来评估稀有矿石类材料的质量。从这一点我们就可以知道，测试和产品质量是紧密联系的。测试事实上包含硬件测试和软件测试两个方面，在本书中特指软件测试(Software Testing)。软件测试在其发展历程中有过不同的定义和观点，了解它们有助于建立正确的软件测试思想。

1.3.1 软件测试的发展历程

迄今为止，软件测试的发展一共经历了五个重要时期。

1) 1957 年之前，以调试为主(Debugging Oriented)

20 世纪 50 年代，英国科学家图灵就给出了软件测试的原始定义：“测试是程序正确性证明的一种极端实验形式”。由于当时的软件规模小、复杂度低、开发过程无序，测试被等同于软件调试，真正意义上的软件测试还未形成。

2) 1957—1978 年，以证明为主(Demonstration Oriented)

1957 年之后，软件测试和软件调试才被区分开来，成为一种发现软件缺陷的独立活动。但是，此时的测试活动往往在代码完成之后进行。相比于软件开发，软件测试的投入非常少，而且缺乏有效的测试方法。20 世纪 50 年代后期到 60 年代，虽然高级语言相继出现，软件复杂度增加，但是相比于硬件系统而言，软件处于次要地位，其正确性仍然主要依赖软件开发人员的水平。这一时期，软件测试理论和方法的发展都很缓慢，测试主要以功能验证为主，测试被看作证明软件正确性的一种方法。1957 年，Charles Baker 对调试和测试进行了区分：调试(Debug)是确保程序做了程序员希望它做的事情，而测试(Testing)是确保程序解决了它该解决的问题。这一区分是软件测试史上的一个重要里程碑，它标志着测试首次具有了独立性。

1972 年，在美国的北卡罗莱纳大学举办了历史上首届正式的软件测试会议，标志着软件测试作为一个学科正式诞生了。软件测试一直与软件工程的发展紧密相关。20 世纪 60 年代中期

之后，随着软件应用数量的急剧增长，软件危机愈演愈烈。为了研究如何通过系统化和工程化的方法应对软件危机，1968 年北大西洋公约组织在联邦德国召开国际会议，会议上正式提出了“软件工程”这一名词，标志着软件工程学科的诞生。随着软件开发在软件工程方法指导下不断正规化，软件测试理论和方法也不断完善。1973 年，William C. Hetzel 整理出版了软件测试的第一本著作 *Program Test Methods*，对测试方法和测试工具进行了论述。1975 年，Goodenough 和 Gerhart 首次提出了软件测试的理论，使得软件测试成为具有理论指导的实践性学科。

3) 1979—1982 年，以破坏为主(Destruction Oriented)

1979 年，Glenford J. Myers(见图 1-7)出版了对软件测试行业影响深远的著作《软件测试的艺术》，他在书中给出了具有开创性意义的软件测试定义：“测试是为了发现错误而执行程序或系统的过程”。这一时期，软件测试的重要意义逐渐被人们所认识，逐渐产生了专业的测试人员，也出现了一些专门的测试工具。同时，人们也意识到，测试不仅是编码之后的一项工作，为了减少改正软件错误的成本，测试工作必须在软件生命周期的前期需求和设计阶段就开始进行，并且需要制定周密的测试计划。受这种软件测试思想的影响，在这一时期，测试往往以破坏性为导向，被看作从软件中寻找错误的过程。

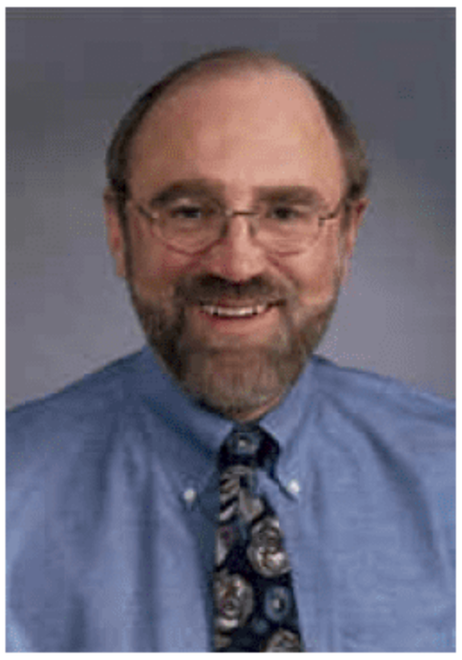


图 1-7　Glenford J. Myers

4) 1983—1987 年，以评估为主(Evaluation Oriented)

20 世纪 80 年代以来，软件行业开始高速发展，软件规模不断增大，复杂度越来越高，并且以各种形式应用于各行各业，深刻影响着人们的生活。因此软件的质量变得越来越重要。软件测试需要运用专门的方法、手段和工具才能满足测试时间和成本的要求，各种实用的软件测试工具应运而生。这一阶段的测试内涵也发生了转变，测试不再是单纯发现错误的过程，而是包含软件质量评价的内容。软件测试成为保障软件质量的重要手段，并且完全融于整个软件开发生命周期。正如 1983 年软件测试先驱 Bill Hetzel 在其著作 *The Complete Guide to Software Testing* 中所描述的软件测试内涵：“测试是以评价程序或系统属性为目标的任何一种活动，是对软件质量的度量”。这一时期，出现了测试领域著名的两个名词：验证(Verification)和确认(Validation)。人们提出了在软件生命周期中通过分析、评审和测试来评估软件产品的理论。

5) 1988 年至今，以预防为主(Prevention Oriented)

2002 年，Rick D. Craig 和 Stefan P. Jaskiel 在 *Systematic Software Testing* 一书中对软件测试进一步阐述为：“测试是为了度量和提高被测软件的质量，对测试软件进行工程设计、实施和维护的整个生命周期过程”，进一步明确了软件测试的目的、价值以及测试活动的系统性。这一时期，预防为主成为软件测试的主流思想之一。STEP(Systematic Test and Evaluation Process)是最早的一个以预防为主的测试生命周期模型，该模型体现了测试与开发的并行性，强调了整个测试的生命周期也由计划、分析、设计、开发、执行和维护组成。也就是说，测试不是在编码完成后才开始介入，而是贯穿于整个软件生命周期。百分之百完美的软件是不存在的，零缺陷是不可能的，所以软件开发过程中测试应当尽早介入，及早发现错误。错误发现得越早，修复的成本越低，产生的风险也越小。

近年来，各种商业化和开源的测试工具大量出现，测试自动化程度不断提高。面向对象测试、面向构件测试、敏捷测试和测试驱动开发等测试思想和方法在实践中得到应用，测试理论