普通高等学校
计算机教育 “十二五”规划教材

近百个二维码，
拓展软件测试知识领域

软件测试
（第2版）

朱少民 ◎ 编著

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

普通高等学校
计算机教育 “十二五”规划教材

软件测试
（第2版）

朱少民 ◎ 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

软件测试 / 朱少民编著. — 2版. — 北京 : 人民邮电出版社, 2016.7
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-41293-5

Ⅰ. ①软… Ⅱ. ①朱… 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(2015)第298402号

内 容 提 要

软件测试是一门新兴的学科，同时又是一门越来越重要的学科。本书首先从软件测试的产生和定义出发，描述了一个完整的软件测试知识体系轮廓，让读者从全局来把握软件测试；然后，针对软件测试的不同知识点展开讨论。

本书在内容组织上力求创新，尽量使软件测试知识具有很好的衔接性和系统性，使需求和设计评审、软件测试用例设计、自动化测试和主要类型的测试活动有机地结合起来，使读者更容易领会如何将测试的方法和技术应用到单元测试、集成测试、系统测试中去。本书提供了丰富的实例和实践要点，特别加强了移动应用 App 的各项测试，从而更好地满足当今软件测试工作的实际需求，使读者掌握测试方法的应用之道和品味测试的最佳实践。

本书条理清晰，语言流畅，通俗易懂，内容丰富、实用，将理论和实践有效地结合起来。本书可作为高等学校的软件工程专业、计算机软件专业和相关专业的教材，成为软件测试工程师的良师益友，也可以用作其他各类软件工程技术人员的参考书。

◆ 编　　著　朱少民
责任编辑　刘　博
责任印制　沈　蓉　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
固安县铭成印刷有限公司印刷

◆ 开本：787×1092　1/16　　拉页：1
印张：19.25　　2016 年 7 月第 2 版
字数：492 千字　　2016 年 7 月河北第 1 次印刷

定价：49.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

第 2 版前言

有时，我不禁会问自己：时间都去哪儿了？《软件测试》第 1 版于 2009 年出版，到现在已经 6 年多了。现在才出第 2 版，更新比较慢，不符合当今敏捷软件开发的快速迭代思想，自己要检讨，要继续不断问自己：时间都去哪儿了？《软件测试》出版这几年还是受到了国内不少大学的欢迎，已重印多次，还成为同济大学“十二五”规划教材，所以笔者有责任及时更新本教材，希望将来每 2~3 年就更新一次，和软件测试技术的发展保持同步。

这几年，最突出的变化是移动应用越来越广泛，如 2014 年移动设备迅速增长，达到 10.6 亿部，是 2013 年的 3 倍，而且每台设备平均安装了 34 个应用程序，平均每天有 20 个左右的移动应用被打开，也就是每天有 200 亿次的移动 App 在运行。移动应用的测试在今天自然也很火，有必要引入到软件测试，所以在第 2 版新增了一章“移动 App 的测试”，不仅涉及其功能测试、自动化测试工具及其应用，而且还详细介绍了移动 App 的专项测试（流量测试、耗电量测试等）、性能测试（侧重内存分析）、安全性测试、针对“闪退”的测试、用户体验测试等。同时，由于本地化和国际化测试已经比较成熟，软件开发平台和语言对本地化和国际化支持越来越好，为了不增加本课程的学时，故将这一章删去，相当于用“移动 App 的测试”这章替换了第 1 版中“本地化测试”那章。除了这项重大改动之外，本书主要还进行了下列一些改动。

（1）引入了敏捷测试，适应当前软件开发模式的变化，但还保留了传统的软件测试。让读者从不同的思维方式来理解软件测试，因此未将“单元测试、集成测试、系统测试、验收测试”看作测试阶段，而是看作测试的不同层次、不同活动。

（2）需求评审也不局限于需求规格说明书的评审，还包括用户故事的评审，如 INVEST 标准。在集成测试中，增加了“持续集成及其测试”的介绍。

（3）第 3 章“测试用例设计”改为“测试分析与设计”，加强测试分析，因为测试分析往往容易被忽视，而实际上，测试分析是测试设计的基础；其次，测试设计也不局限于测试用例的设计。如果不写测试用例，而是开发自动化测试脚本，也需要测试设计。再进一步，在探索式测试中，没有测试用例，也依然需要测试设计。

（4）在功能测试上，不仅加强业务分析，而且扩展整体的分析思路，给出 LOSED 模型，从多个方面去分析，相互补充，更好地确保测试的充分性。

（5）对测试工具进行了更新，删除一些淘汰的工具，增加了一些新出现、更流行的一些测试工具，确保工具的有效性。

（6）增加了 8 个实验，分布在第 2~ 8 章，这些实验覆盖需求评审、测试设计、单元测试、系统功能测试、性能测试、安全性测试、移动应用自动化测试、Windows 应用自动化测试等。当然，教师还可以结合自己学校的特点，安排一些其他实验。

（7）将书中出现的一些链接转换成了二维码，全书二维码近百个，有利于读者更方便、更快地获得相应的阅读材料。

希望通过这次修改，广大教师和学生能够更喜欢本教材。但同时，由于笔者毕竟时间和能力都有限，本教材还会存在一些问题，请大家不吝指正。我自己也会继续努力，不断完善本教材，并尽快推出第3版。这个过程也需要大家的支持，及时提供反馈，为下一版的改进提出宝贵意见，在此表示由衷的感谢！

朱少民

于同济大学美丽的校园

第 1 版前言

在过去的半个世纪，软件获得了空前的发展，逐渐渗透到各个领域，从最早的科学计算、文字处理、数据库管理、银行业务处理到工业自动控制和生产、办公自动化、新闻媒体、通信、汽车、消费电子、娱乐等，软件无处不在，改变了人类的生活与生产方式。随着计算机软件在各行各业的普及应用，人们对软件质量的要求也越来越高，专业化和多样化的特点越来越显著。但同时，我们看到软件产业还不够成熟，软件质量状况不容乐观，软件在运行和使用过程中出现的问题还比较多。例如，2008 年互联网 Web 发展十大失败的事件中，90%的失败都是由软件质量问题造成的，与“宕机”“停机”“崩溃”等一系列严重的质量问题联系在一起。

软件质量一直是软件工程中的一个焦点，成为人们几十年来不断研究、探索的领域。为了改善软件质量，人们不仅从企业文化、软件过程模型、需求工程、设计模式等不同方面来获取有效的方法和最佳的实践，而且开始重视软件测试，在软件测试上有更多的考虑和投入。虽然质量是内建的，但软件测试依旧承担着非常重要的作用。软件测试自身也在发生变化，已经不再只充当门卫的角色——在软件发布之前进行检验，而是正在成为一个持续的反馈机制，贯穿软件开发的整个过程，能尽早地发现问题，降低开发成本，提高软件开发生产力。软件测试人员不再是软件开发的辅助人员，而是软件开发团队的主体之一、积极的参与者。从项目开始的第一天，测试人员就参与项目需求和设计的讨论、评审等各种活动，尽早发现软件需求定义和设计实现上的问题，及时发现软件项目中存在的质量风险。软件开发团队必须尽可能地在交付产品之前控制未来的质量风险，这就必然需要依赖于卓有成效的软件测试。将传统的程序测试的狭义概念扩展到今日业界逐渐认可的、广义的软件测试概念，测试涵盖了需求验证（评审）、设计验证（评审）等活动。软件测试贯穿整个软件生命周期，从需求评审、设计评审开始，就介入软件产品的开发活动或软件项目实施中，和其他开发团队相互协作、相互补充，构成软件生命周期中的有机整体。

软件测试不是一项简单的工作，远比人们所直观想象的要复杂。高效、高质量地完成一个软件系统的测试，涉及的因素很多，也会碰到各种各样的问题，并且要在测试效率和测试风险之间找到最佳平衡点和有效的测试策略，这些都需要测试人员一一克服。要做好软件测试，不仅需要站在客户的角度思考问题，真正理解客户的需求，具有良好的分析能力和创造性的思维能力，完成功能测试和用户界面的测试，而且能理解软件系统的实现机理和各种使用场景，具有扎实的技术功底，通过测试工具完成相应的性能测试、安全性测试、兼容性测试和可靠性测试等更具挑战性的任务。软件测试的主要目的是发现软件中的缺陷，坚持“质量第一”的原则，在实际操作中会遇到一些阻力，需要测试人员去克服。从这些角度看，要成为一个优秀的测试工程师，其实比对成为设计、编程人员的要求还要高，测试人员不仅要体现高超的技术能力，如系统平台设

置、架构设计分析、编程等方面的能力，而且要展示自己的业务分析能力、对客户需求的理解能力和团队沟通协作的能力。

本书在内容上进行了精心组织，从概念到方法，再从方法到实践，满足知识层次和逻辑关系的要求，使课程教学和学习更加自然和有效。例如，先介绍软件测试用例的基本内容和基本方法，然后结合单元测试、功能测试和系统测试来介绍测试用例的具体设计方法，包括白盒方法和黑盒方法。再如，先介绍自动化测试的特点、原理和方法，然后在后面各章结合实际测试需要，引入所需要的测试工具，使方法、工具和测试实践有机地结合起来，使读者更容易理解和掌握工具的使用。

本书全面介绍了软件测试的先进理念和知识体系，演绎了如何使用软件测试的方法、技术和工具，帮助大家尽快成为优秀的测试工程师。首先，在第1章回答了一系列关于软件测试的基本问题。

- 为什么要进行软件测试？
- 软件测试是什么？
- 软件测试学科是如何发展而来的？
- 软件测试带给我们什么？

然后逐步地深入到软件测试领域的各个知识点，包括需求和设计的评审、测试用例设计、自动化测试、缺陷报告、单元测试、功能测试、系统测试、测试计划和管理等。

- 第2章，不仅介绍了各种评审方法，而且从测试人员角度出发，讨论了如何理解需求和设计实现、如何对需求和设计进行评审，包括清晰地描述了需求评审和各类设计评审的标准。
- 第3章，从一个简单的实例引入测试用例，阐述了为什么需要测试用例以及如何从书写、基本设计方法和评审等各个方面保证测试用例的质量，并说明如何组织、使用和维护测试用例。
- 第4章，也是从一些有趣的、简单的实验和一个实实在在的自动化测试的例子，引入软件测试自动化，然后介绍自动化测试的概念、原理、特点和优势，包括代码分析、对象识别和脚本技术。最后，讨论了如何引入自动化测试以及选择测试工具。
- 第5章，重点介绍了单元测试中白盒测试方法和代码评审，包括分支覆盖、条件覆盖法、基本路径、组合覆盖等方法和代码缺陷检查表等。而且，还讨论了集成测试的策略和方法，以及单元测试工具，包括JUnit和微软VSTS等。
- 第6章，重点介绍了功能测试用例的各种设计方法，包括等价类划分、边界值分析、因果图、决策表、功能图和正交试验等方法，并讨论了可用性测试、功能测试执行策略和实践、功能测试工具及其使用。
- 第7章，介绍了国际化和本地化的概念及其关系，详细展示了国际化测试和本地化测试的要求、方法、工具和具体的技巧，包括功能、数据格式、UI、配置、兼容性和翻译等方面的验证。

- 第 8 章，内容丰富，详细讨论了负载测试、压力测试和性能测试等概念及其联系，重点讨论了负载测试和性能测试的方法、技术和工具，包括输入/输出参数、场景设置、结果分析等。然后，介绍了兼容性测试、安全性测试、容错性测试和可靠性测试等。
- 第 9 章，描述了缺陷报告的内容、格式和各种属性，如类型、来源、严重性和优先级等，重点阐述了缺陷生命周期、如何有效地报告和处理缺陷、各种缺陷分析方法及其作用。
- 第 10 章，强调了测试原则，详细介绍了软件测试计划的过程，包括如何制定测试的目标和策略、如何分析测试范围和预估测试的工作量、如何完成资源安排和进度管理等。最后，讨论了测试风险、测试报告和测试管理工具等。

本书特别重视理论与实践相结合，使读者既能领会软件测试的思想和方法，又能将这些方法和技术应用到实际工作中去。因此，本书既适合作为计算机应用、计算机软件、软件工程、软件测试等学科的大学教材，也适合从事软件开发和维护的工程技术人员阅读，包括软件测试人员、开发人员、项目经理和产品经理。

由于作者水平有限，本书不可避免会存在一些错误、不准确以及其他问题，恳请读者见谅并能及时提出宝贵意见。

作　者

2009 年 1 月

目 录

第1章 软件测试概述

TestZilla测试社区

我们匆忙地将众测平台 TestZilla（www.TestZilla.org）发布之后，让同学们去测试。虽然发布之前开发人员已经做了基本的测试，但是同学们还是发现了几十个 Bug，主要包括以下问题。

TestZilla 网站

- ✧ 输入的问题描述格式无法正常显示。
- ✧ Markdown 语法支持不完整。
- ✧ 当用户提交了缺陷后不能正常修改。
- ✧ 本地化缺陷：语言切换功能在产品列表中切换不了。
- ✧ 安全性缺陷：进入账号后可以随意修改密码，并没有邮箱验证等措施。
- ✧ 一个问题提交后再马上新建问题时，编辑框内保存了上一个问题的数据。
- ✧ home page 滚动显示插件、滚动时机问题。

人们在使用一个刚发布的软件时，总是比较容易发现问题，特别是在互联网时代，人们更急于把产品推向市场，即使产品还有比较多的缺陷。无论是知名互联网企业的产品，还是某个创业公司的产品，这种现象比较常见。产品不仅在功能上出现问题，而且在性能、安全性、易用性等各个方面上也会出现问题，例如现在移动 App 应用越来越多，但常常出现闪退（应用崩溃）、兼容性不好、耗流量、耗电等问题。

软件产品形式越来越多，系统越来越复杂，会存在各种各样的问题，而这需要通过软件测试来发现这些问题，并促使这些问题得到修正，从而所发布的软件能够满足质量要求，受到用户的喜欢。软件测试是软件生命周期中最重要的活动之一，从需求评审开始，完成软件定义、设计和实现的验证，全过程地揭示产品质量风险，成为软件质量的守护者，帮助企业获得更强

的市场竞争力和更高的利润。

1.1 一个真实的故事

图 1-1 海军少将 Grace Hopper（1906.12.9 – 1992.1.1）

这是一个真实的故事，故事发生在 1945 年 9 月 9 日一个炎热的下午。当时的机房是一间第一次世界大战时建造的老建筑，没有空调，所有窗户都敞开着。Grace Hopper（程序语言编译器的第一位开发人员，后来成为海军少将，如图 1-1 所示）正领导着一个研究小组夜以继日地工作，研制一台称为“MARK II”的计算机，它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），一台不纯粹的电子计算机。突然，MARK II 死机了。研究人员试了很多次还是无法启动，然后就开始用各种方法找问题，看问题究竟出现在哪里，最后定位到板子 F 第 70 号继电器出错。Hopper 观察这个出错的继电器，惊奇地发现一只飞蛾躺在中间，已经被继电器打死。他小心地用镊子将蛾子夹出来，用透明胶布贴到“事件记录本”中，并注明“第一个发现虫子的实例”，然后计算机又恢复了正常。从此以后，人们将计算机中出现的任何错误戏称为“臭虫”（Bug），而把找寻错误的工作称为“找臭虫”（Debug）。Hopper 当时所用的记录本，连同那只飞蛾，一起被陈列在美国历史博物馆中，如图 1-2 所示。

Bug 故事

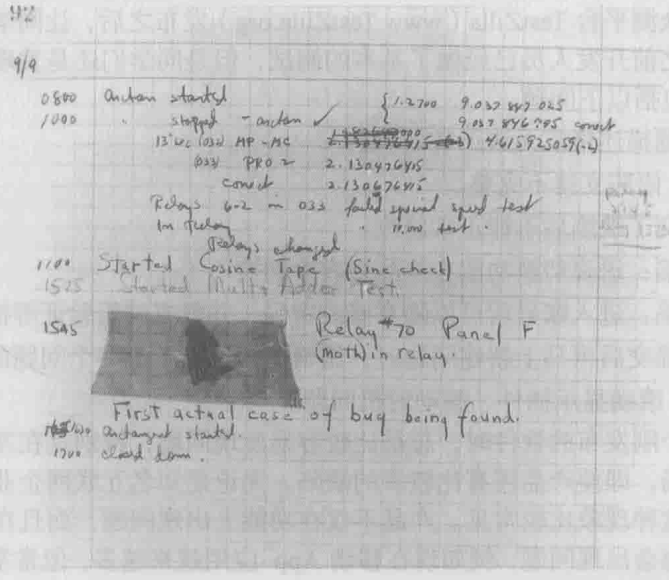

图 1-2 陈列在美国历史博物馆的记录本和飞蛾

（来源：Department of the Navy，Navy Historical Center，Washington，DC ）

这个故事告诉我们，在软件运行之前，要将计算机系统中存在的问题找出来，否则计算机系统可能会在某个时刻不能正常工作，造成更大的危害。从这个故事中，我们也知道软件缺陷

被称为“Bug”的原因；而且知道在什么时候，第一个“Bug”被我们发现。

1.2　为什么要进行软件测试

为什么要进行软件测试？就是因为软件缺陷的存在。因为只有通过测试，才可以发现软件缺陷。也只有发现了缺陷，才可以将软件缺陷从软件产品或软件系统中清理出去。至于为什么会存在软件缺陷，我们将留在下一节来讨论。

软件缺陷的危害有小有大，小的缺陷可能使软件看起来不美观，使用起来不流畅或不方便。而严重的缺陷则可能给用户带来损失，甚至造成生命危险，也可能给软件企业自身带来巨大损失。下面列举的几个例子可以说明软件缺陷所带来的严重危害。美国商务部国家标准和技术研究所（NIST）进行的一项研究表明，软件缺陷每年给美国经济造成的损失高达 595 亿美元。这说明软件中存在的缺陷给我们带来的损失是巨大的，也说明了软件测试的必要性和重要性。

【例一】

苹果推出万众期待的 iPhone 3G 的同时，也推出了一个同步服务器 MobileMe。MobileMe 允许 Mac 和 PC 用户通过一个 Web 界面去同步他们的联系人、日历、电子邮件、照片等内容。但在它推出的第一天便出现了大量的问题——性能缓慢、宕机、用户随机注销等，还有一个致命的问题——整整耗费了一天的时间，同步服务也无法同步日历和全部联系人。就像苹果 CEO Steve Jobs 在一封内部邮件里所写的一样——这不是苹果的“光荣时刻”。后来，苹果修复了那些漏洞，并且承诺所有的 MobileMe 用户可以免费使用 90 天。2008 年，互联网产品宕机的现象非常严重，包括 Twitter 网站频繁出现宕机，Gmail 服务宕机 30 小时等，Twitter 宕机标志 Fail Whale 甚至拥有了其狂热者（Fans）的俱乐部、商店等，如 http://www.zazzle.com/failwhale。

【例二】

当 Facebook 推出创新广告平台 Beacon 的时候，受到了极其严厉的批评。事实证明，Facebook 的用户不喜欢让 Web 上的每个人知道他们的交易记录。例如一个小伙子在某电子商务站点上买了订婚戒指，他的 Facebook 资料里会立刻显示这个交易信息，从而暴露了不该暴露的信息，毁坏了他的订婚惊喜。Facebook 之后在 Beacon 里增加了选项，允许用　　置，不显示相关信息。但是很多不良的影响已经造成了，例如一对夫妇诉讼 Facebook 及提供　　务的合作伙伴。

【例三】

2008 年 8 月，诺基亚公司承认其 Series40 手机平台存在严重缺陷，Series40 手机所使用的旧版 J2ME 中的缺陷使黑客能够远程访问本应受到限制的手机功能，使黑客能够在他人的手机上秘密地安装和激活应用软件。

【例四】

2007 年 10 月 30 日上午 9 点，北京奥运会门票面向境内公众第二阶段预售正式启动。由于瞬间访问数量过大造成网络堵塞，技术系统应对不畅，造成很多申购者无法及时提交申请。为此，票务中心向广大公众表示歉意，并宣布暂停第二阶段门票销售。

【例五】

2007 年，美国 12 架 F-16 战机执行从夏威夷飞往日本的任务中，因计算机系统编码中出现了一个小错误，飞机上的全球定位系统纷纷失灵，导致一架战机折戟沉沙。

【例六】

2003年8月14日发生的美国及加拿大部分地区史上最大停电事故是由软件错误所导致的。SecurityFocus 的数据表明，位于美国俄亥俄州的第一能源（FirstEnergy）公司下属的电力监测与控制管理系统“XA/21”出现的软件错误，是北美大停电的罪魁祸首。根据第一能源公司发言人提供的数据，由于系统中重要的预警部分出现严重故障，负责预警服务的主服务器与备份服务器接连失控，使得错误没有得到及时通报和处理，最终多个重要设备出现故障导致大规模停电。

【例七】

2003年8月11日，“冲击波”计算机病毒首先在美国发作，使美国的政府机关、企业及个人用户的成千上万的计算机受到攻击。随后，“冲击波”蠕虫很快在因特网上广泛传播，结果使十几万台邮件服务器瘫痪，给整个世界范围内的Internet通信带来惨重损失。“冲击波”计算机病毒仅仅利用了微软 Messenger Service 中的一个缺陷，便攻破了计算机安全屏障，可使基于Windows操作系统的计算机崩溃。

【例八】

导航软件Bug使俄罗斯飞船偏离降落地。2003年5月4日，搭乘俄罗斯“联盟-TMA1”载人飞船的国际空间站第七长期考察团的宇航员们返回了地球。但在返回途中，飞船偏离了降落目标地点约460公里。据来自美国国家航空航天局的消息，这是由飞船的导航计算机软件设计中的错误引起的。

【例九】

仅仅由于没有进行有效的集成测试，就导致1999年美国宇航局的火星探测器在试图登陆火星表面时坠毁。原因是当探测器的脚迅速摆开准备着陆时，触发了着地开关，设置了错误的数据位，导致探测器在着陆之前反推器被关闭。

【例十】

爱国者导弹防御系统存在一个致命的软件缺陷，当时钟累计运行超过14小时后，系统的跟踪系统就不准确了。在1999年多哈袭击战中，爱国者导弹防御系统由于连续运行100多个小时，导致了问题的发生。防御系统未能防住飞毛腿导弹，从而造成28名美国士兵死亡。

【例十一】

1996年欧洲航天局阿丽亚娜5型火箭发射后40秒钟火箭爆炸，发射基地2名法国士兵当场死亡，历时9年的航天计划严重受挫，震惊了国际宇航界。爆炸原因在于惯性导航系统软件技术和设计的小失误。

【例十二】

1994年，已大批卖出的英特尔奔腾 CPU 芯片存在浮点运算的缺陷，导致英特尔公司为此付出4.5亿美元的代价。

1.3 软件缺陷的由来

软件缺陷是指软件中存在各种各样的问题，包含了一些偏差、谬误或错误，其主要表现形式是结果出错、功能失效、与用户需求不一致（偏差）等。软件产品或系统中存在的任何一种影响正常运行能力的问题、错误，或者隐藏的功能缺陷或瑕疵，都被认为是软件缺陷。说到底，

软件缺陷会导致软件产品在某种程度上不能满足用户的需要。IEEE 国际标准 729 给出了软件缺陷的定义——软件缺陷就是软件产品中所存在的问题，最终表现为用户所需要的功能没有完全实现，不能满足或不能全部满足用户的需求。

（1）从产品内部看，软件缺陷是软件产品开发或维护过程中所存在的错误、误差等各种问题。

（2）从外部看，软件缺陷是系统所需要实现的某种功能的失效或违背。

软件缺陷反映了软件开发过程中需求分析、功能设计、用户界面设计、编程等环节所隐含的问题。软件缺陷表现的形式有多种，不仅体现在功能的失效方面，而且体现在其他方面，例如表现为：

- ✧ 设计不合理，不是用户所期望的风格、格式等；
- ✧ 部分实现了软件某项功能；
- ✧ 实际结果和预期结果不一致；
- ✧ 系统崩溃，界面混乱；
- ✧ 数据结果不正确，精度不够；
- ✧ 存取时间过长，界面不美观。

由于软件开发人员思维上的主观局限性，且目前开发的软件系统都具有相当的复杂性，决定了在开发过程中出现软件错误是不可避免的。引起软件缺陷的原因比较复杂，来源于方方面面，有软件自身的问题，也有沟通问题，还有技术问题等。主要的因素有：

- ✧ 在需求定义时，用户或产品经理在产品功能上还没有想清楚；
- ✧ 当一个产品在做出来之前，在设想过程中，很难构想得很完美；
- ✧ 需求分析、系统设计时，相关方不能准确表达自己的意见，沟通之间也会存在误解，特别是技术人员和用户、市场人员的沟通存在较大的困难；
- ✧ 沟通不充分，或在多个环节沟通以后信息容易失真，误差会不断放大，真可谓“失之毫厘，谬以千里”；
- ✧ 软件设计规格说明书（functional specification，Spec）中有些功能不可能或无法实现；
- ✧ 系统设计存在的不合理性，难以面面俱到，容易忽视某些质量特性要求；
- ✧ 复杂的程序逻辑处理不当，数据范围的边界考虑不够周全，容易引起程序出错；
- ✧ 算法错误或没有进行优化，从而造成错误、精度不够或性能低下；
- ✧ 软件模块或组件多，接口参数多，配合不好，容易出现不匹配的问题；
- ✧ 异常情况，一时难以想到某些特别的应用场合，如时间同步、大数据量和用户的特别操作等；
- ✧ 其他人为错误，文字写错、数据输入出错、程序敲错等。

1.4　软件测试学科的发展历程

前面的故事告诉我们，任何一个系统都可能存在故障，计算机系统也不例外。有了计算机软件，就可能存在问题；有了问题，就需要测试，一切看起来都是自然而然地发展起来的。实际上，软件测试的发展历程并不一帆风顺，经历了一些波折。特别是在早期阶段，软件测试得不到重视，测试技术发展比较慢，而在最近 20 年发展比较快，成为软件业的热门领域之一，也达到了较高的水准。


测试的历史