



中山大學
SUN YAT-SEN UNIVERSITY

Chapter 2

Basic Concepts of Software Testing

Software Testing: Approaches & Technologies

School of Data & Computer Science, Sun Yat-sen University

Outline

- 2.1 软件缺陷
- 2.2 软件测试概述
 - 软件测试的发展历程
 - 软件测试的目的和意义
 - 软件测试的原则和度量
 - 国内外软件测试行业现状
- 2.3 软件测试的过程和方法
- 2.4 基于软件生命周期的软件测试方法
- 2.5 软件测试的分类与分级



2.1 软件缺陷

- 软件缺陷的概念

- Software Defect

- ✧ A software defect is any problem/disfigurement/limitation in product design & development
 - ✧ A software bug is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. (Wiki)
 - Feature or function can't work
 - Unreasonable design
 - Partly realization in function
 - Data error
 - Run error
 - Limitation in features
 - Difference between actual results and expected results
 - Unfriendly UI
 - Low performance
 - Others.

2.1 软件缺陷

- 软件缺陷的概念

- 什么是软件缺陷

- ◇ 任何计算机程序或系统中的问题、与产品设计书的不一致性、不能满足用户的需求等都会形成软件缺陷。

- 软件缺陷将导致软件故障

- ◇ 计算机系统的规模和复杂性急剧增加，计算机软件的数量快速膨胀。与此同时，计算机出现故障引起系统失效的可能性也逐渐增大。随着计算机硬件技术的进步，元器件可靠性的提高，硬件设计和验证技术的成熟，硬件质量冗余度的提高，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。

2.1 软件缺陷

- 软件缺陷的概念

- IEEE 729-1983, GB/T 11457-89

- ✧ 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、毛病等各种问题；
 - ✧ 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。

2.1 软件缺陷

- 软件缺陷的概念

- Ron Patton (Software Testing, 2005):

- ✧ 至少满足以下5个规则之一，才称为发生了一个软件缺陷

- (1) 软件未实现产品说明书要求的功能

- (2) 软件出现了产品说明书指明不应该出现的错误

- (3) 软件实现了产品说明书未提到的功能

- (4) 软件未实现产品说明书虽未明确提及但应该实现的目标

- (5) 软件难以理解，不易使用，运行缓慢或者—从测试员的角度看—最终用户会认为不好

- How to patch a software defect

- ✧ Better management

- ✧ Different team organizations

- ✧ Better languages & tools

- ✧ Uniform coding conventions

2.1 软件缺陷

- 软件缺陷的概念

- 软件缺陷的主要类型/现象

- ✧ 约定的功能、特性没有实现或只是部分实现
 - ✧ 设计不合理
 - ✧ 实际结果和预期结果不一致
 - ✧ 运行出错，包括运行中断、系统崩溃、界面混乱
 - ✧ 数据结果不正确、精度不够
 - ✧ 用户不能接受的其他问题，如存取时间过长、界面不美观

2.1 软件缺陷

- **Spectacular Software Failures**

- NASA's Mars lander
 - ✧ September 1999, crashed due to a units integration fault
- THERAC-25 radiation machine
 - ✧ Poor testing of safety-critical software can cost lives : 3 patients were killed
- Ariane 5 explosion
 - ✧ exception-handling bug : forced self destruct on maiden flight (64-bit to 16-bit conversion: about 370 million \$ lost)
- Intel's Pentium FDIV fault
 - ✧ Public relations nightmare
- Northeast Blackout of 2003
 - ✧ The alarm system in the energy management system failed due to a software error and operators were not informed of the power overload in the system.
 - ✧ Financial losses of \$6 Billion USD

2.1 软件缺陷

- 软件缺陷的5级分级体系

- Fatal

- ✧ 致命的错误。缺陷造成系统或应用程序崩溃、死机、系统挂起，或造成数据丢失、主要功能完全丧失等。

- Critical

- ✧ 严重错误。缺陷导致某些功能或特性没有实现，主要功能部分丧失，次要功能完全丧失，或发出致命的错误声明。

- Major

- ✧ 不太严重的错误。缺陷虽然不影响系统的使用，但没有很好地实现系统功能，没有达到预期效果，例如次要功能丧失，提示信息不太准确，用户界面差，操作时间长等。

2.1 软件缺陷

- 软件缺陷的5级分级体系

- Minor

- ✧ 一些小问题。缺陷对系统功能几乎没有影响，产品及属性仍可使用。

- Suggestion

- ✧ 建议改进。



2.1 软件缺陷

- 软件缺陷状态

缺陷状态	描 述
激活或打开 Active or Open	缺陷还没有解决，在等待处理中，如新报告的缺陷
已修正或修复 Fixed or Resolved	缺陷已被开发人员检查、修复过，通过单元测试，但还没有被测试人员验证
关闭或非激活 Closed or Inactive	缺陷经测试人员验证后，确认已经修复、或不存在之后的状态。
重新打开 Reopen	缺陷经测试人员验证后还依然存在，等待开发人员进一步修复
推迟 Deferred	这个缺陷可以在下一个版本中解决
保留 On hold	由于技术原因或第三方软件的缺陷，开发人员不能修复的缺陷

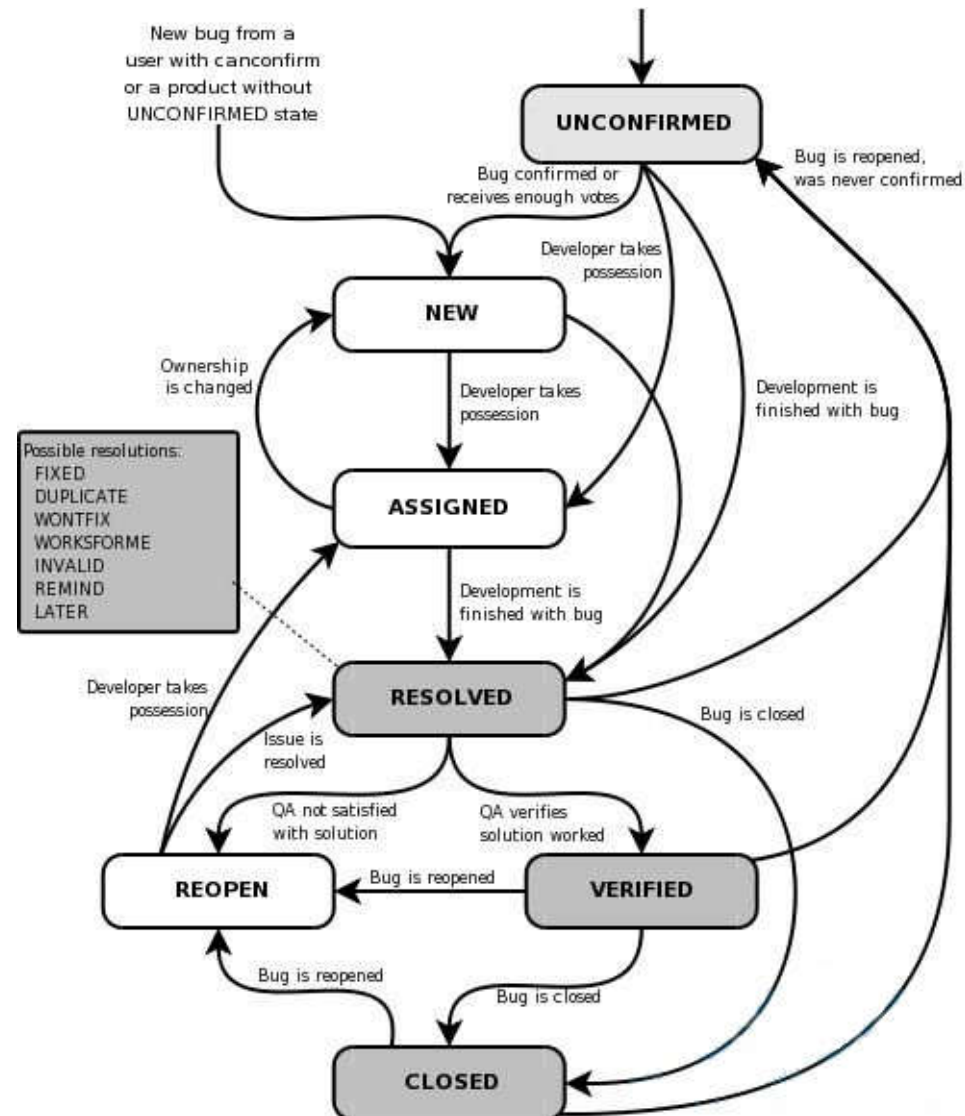
2.1 软件缺陷

- 软件缺陷状态

缺陷状态	描 述
不能重现 Can not duplicate	开发者不能复现这个缺陷，需要测试人员检查缺陷复现的步骤
需要更多信息 Need more info	开发能复现这个缺陷，但开发者需要一些信息，如：缺陷的日志文件，图片等
重复 Duplicate	这个缺陷已经被其他的软件测试人员发现
不是缺陷 Notabug/Reject	这个问题不是软件缺陷
需要修改软件规格说明书 Spec modified	由于软件规格说明书对软件设计的要求，软件开发人员无法修复这个缺陷，必须修改软件规格说明书。

2.1 软件缺陷

- 软件缺陷的生命周期



2.1 软件缺陷

- 软件缺陷的产生原因

- 团队协作问题

- ✧ 沟通不准确或不充分产生的误解

- 技术问题

- ✧ 算法错误，语法错误，计算和精度问题

- ✧ 接口参数匹配问题

- 软件本身问题

- ✧ 文档错误或不足、用户使用场合 (User Scenario) 不清晰

- ✧ 时间上不协调、或不一致

- ✧ 系统的自我恢复或数据的异地备份、灾难性恢复问题

- ✧ 软件可靠性缺少度量的标准

- ✧ 软件难以维护、不易升级

2.1 软件缺陷

- 软件缺陷的术语

- 常用术语

- ✧ Error 错误
 - ✧ Defect/Bug 缺陷
 - ✧ Fault 故障
 - ✧ Failure 失效
 - ✧ Anomaly 异常
 - ✧ Variance 偏差
 - ✧ Incident (小)毛病
 - ✧ Inconsistency 矛盾/不一致
 - ✧ Problem 问题

- 软件失效过程

- ✧ Software Error -> Software Fault -> Software Failure

2.1 软件缺陷

- **Software Faults, Errors & Failures**

- Error (错误：病因)

- ✧ An incorrect internal state that is the manifestation of some fault
 - E.g., high blood pressure, irregular heartbeat, bacteria in the blood stream

- Fault (故障：有病)

- ✧ A static defect in the software
 - Like design mistakes in hardware
 - E.g., the ailment (疾病)

- Failure (失效：故障的**动态**表现)

- ✧ External, incorrect behavior with respect to the requirements or other description of the expected behavior
 - E.g., a list of symptoms (一系列症状)

- The word “Bug” is used informally.

- ✧ A software bug is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

2.1 软件缺陷

- **Software Faults, Errors &**

Fault: Should start searching at 0, not 1

```
public static int numZero (int [ ] arr)
{ // Effects: If arr is null throw NullPointerException
  // else return the number of occurrences of 0 in arr
  int count = 0;
  for (int i = 1; i < arr.length; i++)
  {
    if (arr [ i ] == 0)
    {
      count++;
    }
  }
  return count;
}
```

Test 1
[2, 7, 0]
Expected: 1
Actual: 1

Error: i is 1, not 0, on the first iteration
Failure: none

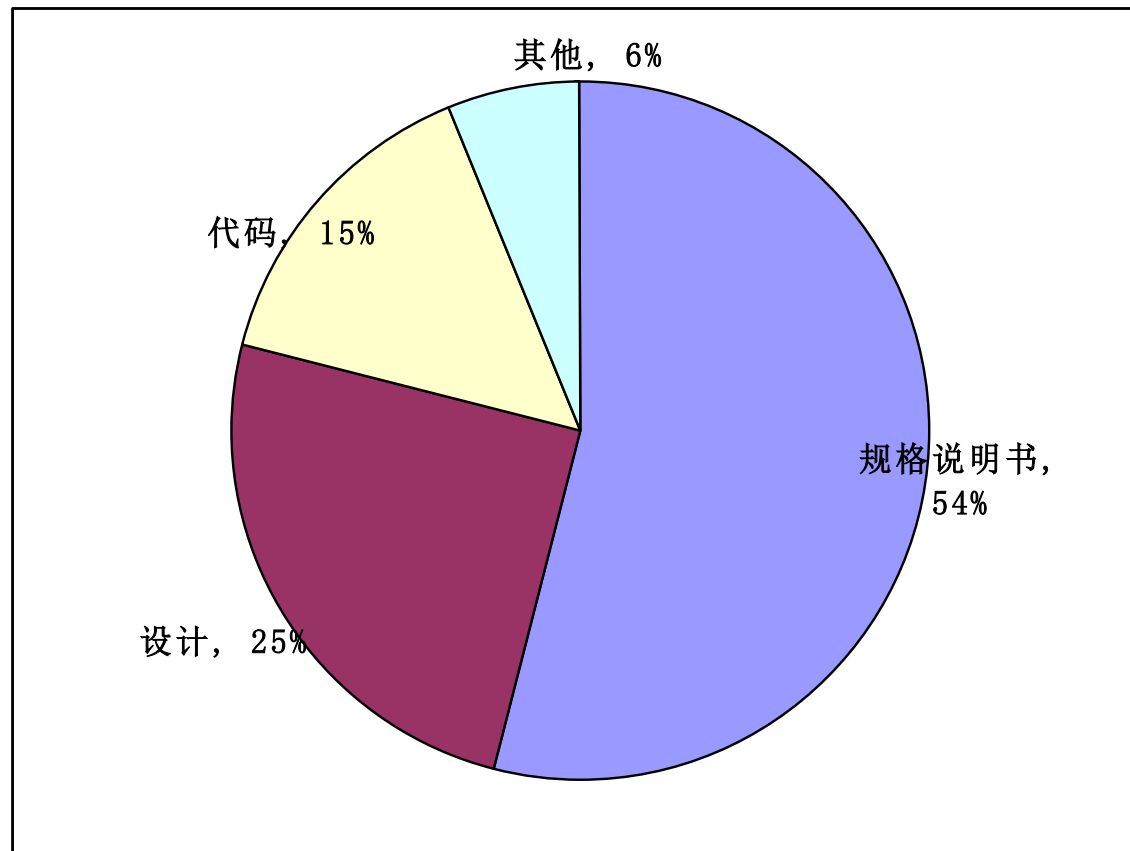
Test 2
[0, 2, 7]
Expected: 1
Actual: 0

Error: i is 1, not 0, on the first iteration
Error propagates to the variable count
Failure: count is 0 at the return statement

— An **error** may occur because of mistyping

2.1 软件缺陷

- 软件缺陷的构成



2.1 软件缺陷

- 规格说明书缺陷的主要原因

- 需求分析阶段的沟通障碍

- ✧ 产品功能：软件开发人员和非计算机专业用户对要开发的产品的功能理解不一致

- ✧ 产品特性：由于软件产品还没有设计、开发，对于产品的表现只能凭经验去估计和想象，有些特性还不够清晰

- 用户的需求变化

- ✧ 用户的需求总是在不断变化的，这些变化如果没有在产品规格说明书中得到正确的最终描述，会引起说明书上下文矛盾

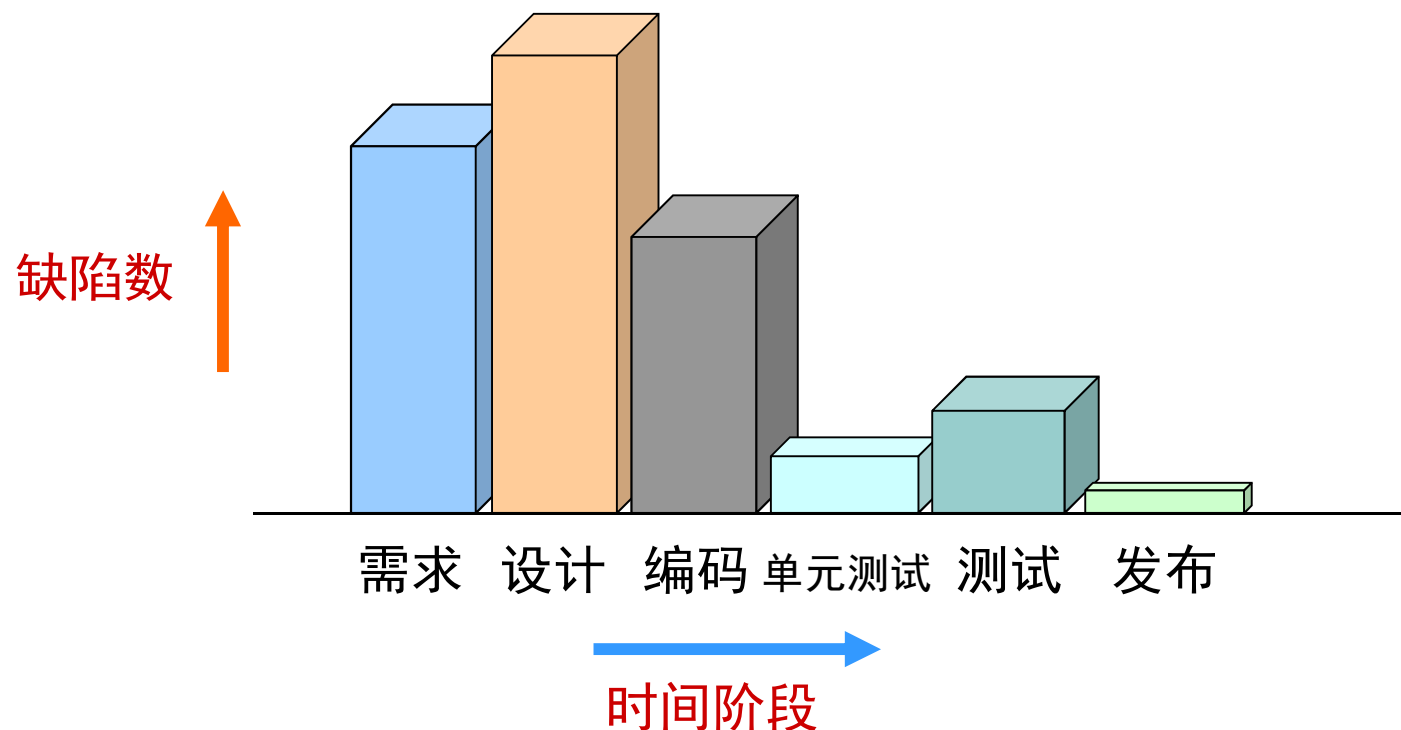
- 主观因素的影响

- ✧ 对规格说明书重视不够，在设计和写作上投入不足

- ✧ 经常只有设计师或项目经理才会得到比较全面的产品规格信息

2.1 软件缺陷

- 软件缺陷在不同阶段的分布

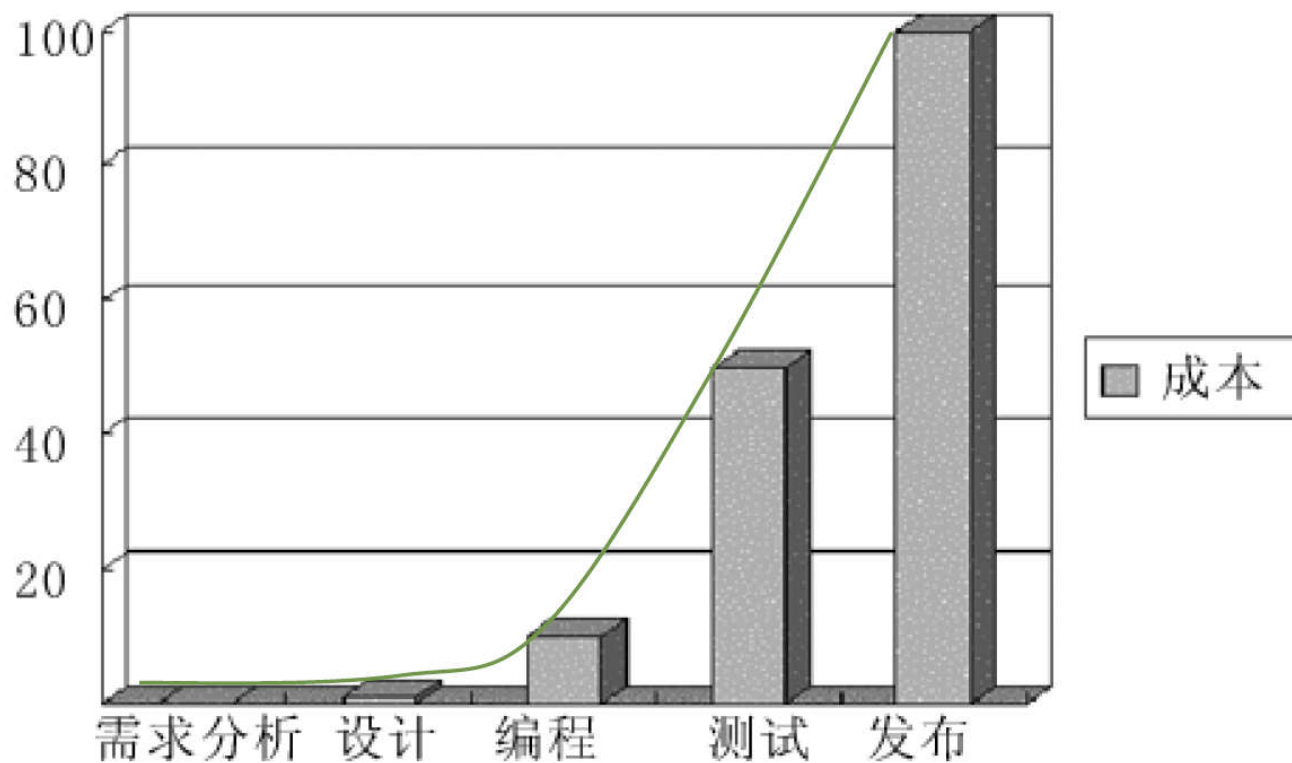


— 早期缺陷发现：测试前 70%-90%

✧ 在真正的程序测试之前，通过审查、评审会可以发现更多的缺陷。

2.1 软件缺陷

- 软件缺陷修复成本



- 未能在早期发现并及时修复的软件缺陷，将在后期导致极大的修复成本。

2.2 软件测试概述

2.2.1 软件测试的发展历程

- 起步阶段
 - 20世纪50-60年代
 - ✧ 英国计算机科学家图灵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。
 - ✧ 20世纪60年代 (软件工程建立前)：软件测试的目的是为了表明程序代码的正确性。
 - 20世纪70年代
 - ✧ 随着计算机硬件技术的进步与成熟，软件在整个系统中所占的地位越来越重要，软件规模和复杂性大大增加，软件可靠性面临前所未有的危机，给软件测试工作带来了挑战。
 - ✧ 测试的意义逐渐被人们认识，软件测试的研究开始受到重视，这是软件测试技术发展最活跃的时期。

2.2 软件测试概述

2.2.1 软件测试的发展历程

- 起步阶段

- 20世纪70年代 (续)

- ✧ 1972年首届软件测试正式会议在北卡罗来纳大学举行。
 - ✧ 1975年 *J. B. Goodenough* 和 *S. L. Gerhart* 发表研究论文 *Towards a Theory of Test Data Selection* (测试数据选择的原理), 首次提出了软件测试理论, 软件测试被确定作为一种研究方向, 软件测试这一实践性很强的学科被提高到理论的高度。
 - ✧ 1979年, *G. J. Myers* 在首版《软件测试艺术》一书中指出: 测试是为发现错误而执行的一个程序或者系统的过程。

2.2 软件测试概述

2.2.1 软件测试的发展历程

- 软件质量标准阶段
 - 20世纪80年代早期开始“软件质量管理”
 - ✧ 软件测试的定义发生了改变，测试不单纯是一个发现错误的过程，而且包含了软件质量评价的内容。
 - ✧ 各类标准被制定出来。
 - ✧ 1983年，*Bill Hetzel* 在《软件测试完全指南》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。

2.2 软件测试概述

2.2.1 软件测试的发展历程

- 20世纪90年代，测试工具开始盛行
 - 1996年开始提出：
 - ✧ 测试能力成熟度 TCMM (Testing Capability Maturity Model)
 - ✧ 测试支持度 TSM (Testability Support Model)
 - ✧ 测试成熟度 TMM (Testing Maturity Model)
- 21世纪
 - 测试的生命周期理论
 - ✧ 2002年，*Rick D. Craig & Stefan P. Jaskiel* 在 *Systematic Software Testing* (系统的软件测试) 一书中对软件测试做了进一步定义：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。

2.2 软件测试概述

2.2.1 软件测试的发展历程

- Testing in the 21st Century
 - Software defines behavior
 - ✧ network routers, finance, switching networks, other infrastructure
 - Today's software market
 - ✧ much bigger, more competitive, more users
 - Embedded Control Applications
 - ✧ airplanes, air traffic control
 - ✧ spaceships
 - ✧ watches
 - ✧ ovens
 - ✧ remote controllers
 - Agile processes put increased pressure on testers
 - ✧ Programmers must unit test
 - ✧ Tests are key to functional requirements
 - Software is a skin that surrounds our civilization

2.2 软件测试概述

2.2.1 软件测试的发展历程

- Testing in the 21st Century
 - More safety critical, real-time software
 - Embedded software is ubiquitous ... check your pockets
 - Enterprise applications means bigger programs, more users
 - Paradoxically, free software increases our expectations !
 - Security is now all about software faults
 - ✧ Secure software is reliable software
 - The web offers a new deployment platform
 - ✧ Very competitive and very available to more users
 - ✧ Web apps are distributed
 - ✧ Web apps must be highly reliable

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- 软件测试的概念

- IEEE, 1983

- ✧ 软件测试是一个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。

- 软件测试的目的

- ✧ 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。寻找软件故障是软件测试的目的

- 软件测试的动机

- ✧ 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。

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- 软件测试的目的

- *G. J. Myers*

- ✧ 软件测试是为了发现错误而执行程序的过程
 - ✧ 一个好的测试能够在第一时间发现程序中存在的错误
 - ✧ 一个好的测试是发现了至今尚未发现的错误的测试。

- 软件测试检查由开发小组开发的软件，测定软件的状态，而不是决定该软件是否能成为产品。

- ✧ 从用户的角度出发，希望通过软件测试暴露软件中隐藏的错误和缺陷，以考虑是否可接受该产品。
 - ✧ 从软件开发者的角度出发，希望测试成为表明软件产品中不存在错误的过程，验证该软件已正确地实现了用户的要求，确立人们对软件质量的信心。

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- 软件测试的目的

- 综合

- ✧ 软件测试以最少的时间和人力，系统地发现软件中潜在的各种错误和缺陷。
 - ✧ 软件测试证明软件的功能和性能与需求说明相符合。
 - ✧ 软件测试收集到的测试数据为软件可靠性分析提供了依据。
 - ✧ 软件测试不能表明软件中不存在错误，它只能说明软件中存在错误。

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- Testing Goals Based on Test Process Maturity
 - Level 0
 - ✧ There's no difference between testing and debugging
 - Testing is the same as debugging
 - Does not distinguish between incorrect behavior and mistakes in the program
 - Does not help develop software that is reliable or safe

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- Testing Goals Based on Test Process Maturity
 - Level 1
 - ✧ The purpose of testing is to show correctness
 - Purpose is to show correctness
 - Correctness is impossible to achieve
 - What do we know if no failures?
 - Good software or bad tests?
 - Test engineers have no:
 - Strict goal
 - Real stopping rule
 - Formal test technique
 - Test managers are powerless

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- Testing Goals Based on Test Process Maturity
 - Level 2
 - ✧ The purpose of testing is to show that the software doesn't work
 - Purpose is to show failures
 - Looking for failures is a negative activity
 - Puts testers and developers into an adversarial relationship
 - What if there are no failures?

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- Testing Goals Based on Test Process Maturity
 - Level 3
 - ✧ The purpose of testing is not to prove anything specific, but to reduce the risk of using the software
 - Testing can only show the presence of failures
 - Whenever we use software, we incur some risk
 - Risk may be small and consequences unimportant
 - Risk may be great and consequences catastrophic
 - Testers and developers cooperate to reduce risk

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- Testing Goals Based on Test Process Maturity
 - Level 4
 - ✧ Testing is a mental discipline (智力训练) that helps all IT professionals develop higher quality software
 - Testing is the only one way to increase quality
 - Test engineers can become technical leaders of the project
 - Primary responsibility to measure and improve software quality
 - Their expertise should help the developers

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- 软件测试的重要性
 - 软件测试是在软件投入运行前，对软件需求，设计和编码的最终复审，是软件质量保证的关键步骤。
 - 软件设计与编码过程是引入错误的过程，而软件测试是通过测试排除软件错误的过程。
 - 测试在软件开发中占有重要地位
 - ✧ 测试成本占开发成本的 30%-50%
 - 软件测试在软件开发过程中的具体作用：
 - ✧ 确认软件产品存在那些属性和缺乏那些属性
 - ✧ 验证需求是否正确，确认最终产品符合用户真正的要求
 - ✧ 对专门的过程进行分类测试，评价产品的合理性

2.2 软件测试概述

2.2.2 软件测试的目的和意义

- 软件测试的重要性
 - 进入20世纪90年代后，计算机技术日趋成熟，软件应用范围逐步扩大，软件规模和复杂性急剧增加，与此同时，计算机出现故障引起系统失效的可能性也逐渐增加。由于计算机硬件技术的进步，元器件可靠性的提高，硬件设计和验证技术的成熟，硬件故障相对显得次要，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。
 - 软件测试仍然是软件工程发展较为薄弱的一个方面。不仅测试理论，已有的测试方法也不能满足当前软件开发的实际需求。

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件测试的公理
 - 软件测试不能表明被测软件中不存在错误，它只能证明被测软件中存在错误。
 - 软件测试的两个作用是：确定被测程序中存在缺陷；判断该程序实际是否可用。
 - 软件测试最困难的问题之一是：何时停止测试？
 - 自己不能测试自己的程序。
 - 当一个软件被测出的缺陷数目增加时，该软件中存在更多的未被发现的缺陷的概率也随之增加。
 - 一个好的测试用例应当是一个对以前未被发现的缺陷有高发现率用例，而不是一个表明被测程序工作正确的用例。

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件测试的公理 (续)
 - 测试用例要兼顾有效和无效的输入。
 - 每个测试用例的必备部分是描述预期的输出。
 - 测试的目标从一开始就确定。



2.2 软件测试概述

2.2.3 软件测试原则和度量

- 足够好原则
 - 一种权衡投入/产出比的原则
 - 不充分的测试是不负责任的；过分的测试是一种资源的浪费，同样也是一种不负责任的表现。
 - ✧ 制定最低测试通过标准和测试内容

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 木桶原理
 - 在软件产品生产方面就是全面质量管理 (TQM) 的概念。
 - 产品质量的关键因素是分析、设计和实现，测试应该是融于其中的补充检查手段，其他管理、支持、甚至文化因素也会影响最终产品的质量。
 - 测试是提高产品质量的必要条件，也是提高产品质量最直接、最快捷的手段，但决不是一种根本手段。将提高产品质量的砝码全部押在测试上是完全不可取的。

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件缺陷发现的80-20原理
 - 一般情况下，在分析、设计、实现阶段的复审和测试工作能够发现和避免80%的缺陷，而系统测试又能找出其余缺陷中的80%，最后的4%的缺陷可能只有在用户的大范围、长时间使用后才会展露出来。
 - 测试只能保证尽可能多地发现错误，无法保证能够发现所有的错误。

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件测试的一般原则

- ✧ 所有测试的标准都是建立在用户需求之上。
- ✧ 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。
- ✧ 事先定义好产品的质量标准，只有有了质量标准，才能根据测试的结果，对产品的质量进行分析和评估。
- ✧ 软件项目启动就是软件测试的开始。“尽早地和不断地进行软件测试”，测试要贯穿整个软件开发生命周期。
- ✧ 在时间和资源的限制下，穷举测试是不可能的。即使一个大小适度的程序，其路径排列的数量也非常大，因此，在测试中不可能运行路径的每一种组合。
- ✧ 第三方进行测试会更客观，更有效。

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件测试的一般原则 (续)

- ✧ 准备好测试计划，排除测试的随意性。
- ✧ 测试用例要根据测试的目的，采用相应的方法去设计，从而提高测试的效率，更多地发现错误，提高程序的可靠性。
- ✧ 测试设计决定了测试的有效性和效率，测试工具只能提高测试效率。
- ✧ 对发现错误较多的程序段，应进行更深入的测试。一般来说，一段程序中已发现的错误数越多，其中存在的错误概率也就越大。
- ✧ 妥善保存一切测试过程文档 (测试计划、测试用例、测试报告等)

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件测试的质量度量
 - 目的
 - ✧ 改进软件测试质量，提高测试效率，改进测试过程的有效性。
 - 难度
 - ✧ 软件产品的质量不能直接反映到软件测试的效果，只能转移到对软件测试过程的度量以及对测试产出物的度量。
 - 关键
 - ✧ 关键是对软件测试人员工作的质量度量。
 - ✧ 测试人员是测试过程的核心人物，测试人员的工作质量会极大地影响测试的质量，以及产品的质量评价。
 - ✧ 对测试人员的工作评价一般由测试经理或项目经理、质量保证人员以及开发人员这三类人员进行综合考核或评判。

2.2 软件测试概述

2.2.3 软件测试原则和度量

-

指标名称	定义	度量范围
工作量偏差	$(\text{实际工作量} - \text{计划工作量}) / \text{计划工作量} * 100\%$	进度
测试执行率	$(\text{实际执行的测试用例数} / \text{测试用例总数}) * 100\%$	测试进度
测试通过率	$(\text{执行通过的测试用例数} / \text{测试用例总数}) * 100\%$	开发质量
需求（测试用例）覆盖率	$(\text{已设计测试用例的需求数} / \text{需求总数}) * 100\%$	测试设计质量
需求通过率	$(\text{已测试通过的需求数} / \text{需求总数}) * 100\%$	进度
测试用例命中率	$(\text{缺陷总数} / \text{测试用例数}) * 100\%$	测试用例质量
二次故障率	$(\text{Reopen的缺陷} / \text{缺陷总数}) * 100\%$	开发质量
NG率	$(\text{验证不通过的缺陷} / \text{缺陷总数}) * 100\%$	开发质量
缺陷有效率	$(\text{无效的缺陷} / \text{缺陷总数}) * 100\%$	测试
缺陷修复率	$(\text{已解决的缺陷} / \text{缺陷总数}) * 100\%$	开发
缺陷生存周期	缺陷从提交到关闭的平均时间	开发、测试
缺陷修复的平均时长	缺陷从提交到修复的平均时间	开发
缺陷关闭的平均时长	缺陷从修复到关闭的平均时间	测试
缺陷探测率	$(\text{测试者发现的缺陷数} / (\text{测试者发现的缺陷} + \text{客户发现的缺陷})) * 100\%$	测试质量

2.2 软件测试概述

2.2.3 软件测试原则和度量

- 软件测试的衡量标准
 - 多
 - ✧ 能够找到尽可能多的、以至于所有的软件缺陷
 - 快
 - ✧ 能够尽可能早地发现最严重的软件缺陷
 - 好
 - ✧ 找到的软件缺陷是关键的、用户最关心的
 - ✧ 找到软件缺陷后能够重现找到的缺陷，并为修正缺陷提供尽可能多的信息
 - 省
 - ✧ 能够用最少的时间、人力和资源发现软件缺陷
 - ✧ 测试的过程和数据可以重用

2.2 软件测试概述

2.2.4 软件测试行业

- 国外软件测试行业的情况
 - 软件终端产业比较发达的国家 (特别是美国), 软件测试已经发展成为一个独立的产业, 体现在:
 - ✧ 软件测试在软件生产过程中占有重要地位, 软件测试占了软件企业的 40% 以上的工作量。
 - ✧ 软件测试人员和软件开发人员之比平均在 1:1 以上
 - 微软达到 1.5:1 至 2.5:1。
 - ✧ 软件测试费用占整体项目开发费用的 30%-50%。
 - ✧ 对于高可靠、高安全的软件, 测试费用相当于软件开发费用的3至5倍。
 - ✧ 软件测试理论研究蓬勃发展。
 - ✧ 软件测试市场繁荣, MI、Compuware、Rational 等机构出品的测试工具在国际市场占有领导地位。



2.2 软件测试概述

2.2.4 软件测试行业

- 国内软件测试行业的情况
 - 目前国内软件业的软件产品测试一般有下列几种形式：
 - ✧ 软件企业内部进行的功能性测试
 - ✧ 用户进行的系统验收测试
 - ✧ 第三方测试：专业软件测试人员运用一定的测试工具对软件的质量进行检测

2.2 软件测试概述

2.2.4 软件测试行业

- 国内软件测试行业的情况 (续)

- 整体状况

- ✧ 重开发、轻测试导致国内软件开发企业在软件测试上的投入远远低于软件开发上的投入，更远远低于软件产业发达国家在软件测试上的投入。
 - ✧ 软件测试人员和软件开发人员之比平均在 1:3 – 1:8 之间。
 - ✧ 软件测试人才的培养水平较发达国家以及我国软件产业提升状态严重滞后，这就形成了软件测试人才的供给严重不足。
 - ✧ 缺乏对软件产品化测试技术的研究。
 - ✧ 测试服务没有形成足够的规模。。。

2.2 软件测试概述

2.2.4 软件测试行业

- 国内软件测试行业的情况 (续)
 - 行业需求
 - ✧ 软件外包中对测试环节的强化，直接导致了软件外包企业对测试人才的大量需求。
 - ✧ 几乎所有的软件企业均存在不同程度的测试人才缺口，软件测试工程师已成为了亟待补充的关键技术工种之一。

2.2 软件测试概述

2.2.4 软件测试行业

- 软件测试职业
 - 软件测试职业分类
 - ◇ 软件测试工具开发工程师
 - 负责编写测试工具代码，并利用测试工具对软件进行测试
 - 开发测试工具为软件测试工程师服务
 - ◇ 软件测试工程师
 - 负责理解产品的功能要求，然后对其进行测试，检查软件是否存在错误，决定软件是否具有稳定性，完成相应的测试计划和测试用例

2.2 软件测试概述

2.2.4 软件测试行业

- 软件测试职业

- 一般软件测试团队人员分类

- ✧ 测试经理：主要负责人员的招聘、培训和管理，以及资源调配、测试方法改进等
 - ✧ 实验室管理人员：设置、配置和维护实验室的测试环境，如服务器和网络环境等
 - ✧ 测试配置管理人员：审查流程，并提出改进流程的建议；建立测试文档所需的各种模板，进行测试的配置管理，检查软件缺陷描述及其他测试报告的质量等
 - ✧ 测试组长：测试技术业务专家，负责项目管理、测试计划制订、项目文档审查、测试用例设计和审查、任务安排，以及和项目经理、开发组长沟通等

2.2 软件测试概述

2.2.4 软件测试行业

- 软件测试职业
 - 一般软件测试团队人员分类 (续)
 - ✧ 一般 (初级) 测试工程师: 编写、执行测试用例, 编制有关的测试文档或开展其它相关的测试任务

2.2 软件测试概述

2.2.4 软件测试行业

- 软件测试职业
 - 软件测试工程师分类：
 - ✧ 资格分级
 - 初级测试工程师
 - 测试工程师
 - 高级测试工程师
 - ✧ 业务归口
 - 自动化测试工程师
 - 系统测试工程师
 - 架构工程师

Thank you!

