软件质量保证与测试

第2讲　软件测试概述

内容提要

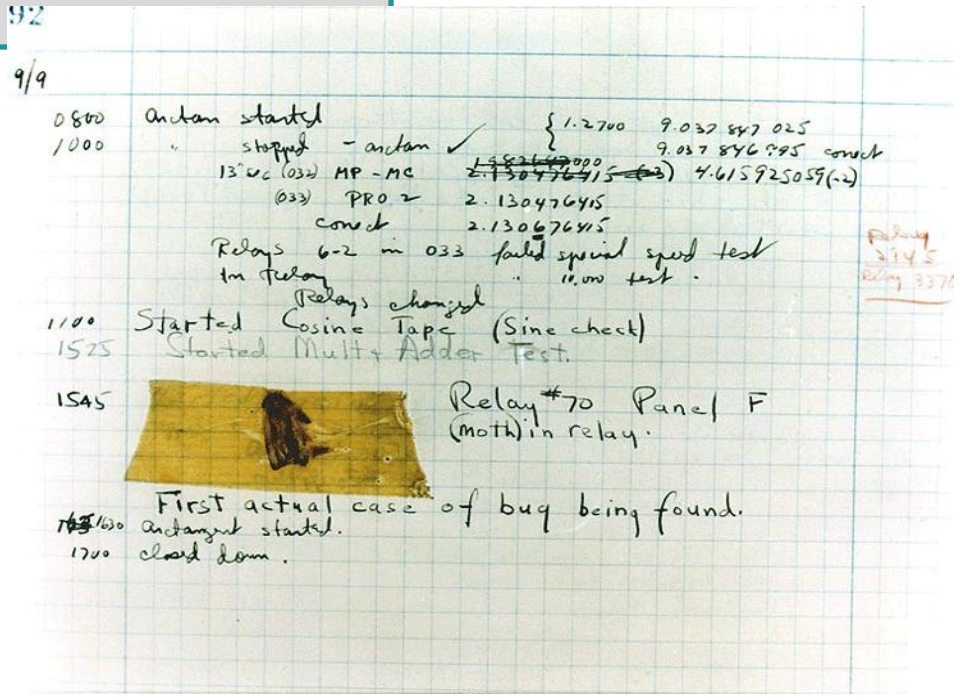
- 1.1 软件测试的意义
- 1.2 软件测试的定义
- 1.3 软件测试的分类
- 1.4 软件测试的过程
- 1.5 小结

内容提要

- 1.1 软件测试的意义
- 1.2 软件测试的定义
- 1.3 软件测试的分类
- 1.4 软件测试的过程
- 1.5 小结

1.1 软件测试的意义

知道”Bug”是怎么来的吗？



软件测试的意义（续）

软件缺陷案例时有发生

正在加载 Gmail

软件测试的意义（续）

- 波音飞机的灾难 （摘自“软件质量报道”公众号）
- 人造陨石坑缺陷
- 迪斯尼狮子王缺陷
- 英特尔浮点除法缺陷
- 程序员的千年虫问题
- Windows的输入法漏洞
- 爱国者导弹缺陷

软件测试的意义（续）

IEEE729对软件缺陷的定义：

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。

- 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
- 从产品外部看：软件缺陷是系统所需要实现的某种功能的失效或违背。

软件测试的意义（续）

软件缺陷的表现形式：

- 软件没有实现需求规格说明所要求的功能模块
- 软件中出现了需求规格说明指明不应该出现的错误
- 软件实现了需求规格说明未提及的功能模块
- 软件没有实现需求规格说明虽未明确提及但应该实现的目标
- 软件难以理解，不容易使用，运行缓慢，或从测试员的角度看，最终用户会认为不好。

- 检查有效输入下的基本功能
- 检查相关性能指标

软件测试的意义（续）

软件缺陷的表现形式：

- 软件没有实现需求规格说明所要求的功能模块
- 软件中出现了需求规格说明指明不应该出现的错误
- 软件实现了需求规格说明没有提到的功能模块
- 软件没…及但应该实现…
- 软件难…从测试员的角…

检查软件容错性：

- 检查异常情况
- 检查无效用户输入的识别能力
- 检查无用用户输入的处理能力

软件测试的意义（续）

软件缺陷的表现形式：

- 软件没有实现需求规格说明所要求的功能模块
- 软件中出现了需求规格说明指明不应该出现的错误
- 软件实现了需求规格说明没有提到的功能模块
- 软件没有实现虽然需求规格说明没有明确提及但应该实现的
- 软件难以

或从测试员的角度

- 测试很难发现
- 开发人员主动避免

软件测试的意义（续）

软件缺陷的表现形式：

- 软件没有实现需求规格说明所要求的功能模块
- 软件中出现了需求规格说明指明不应该出现的错误
- 软件实现了需求规格说明没有提到的功能模块
- 软件没有实现虽然需求规格说明没有明确提及但应该实现的目标
- 软件难以理解、不容易使用，运行缓慢，或从测试员的角度看最终用户会认为不好。

隐含的质量需求

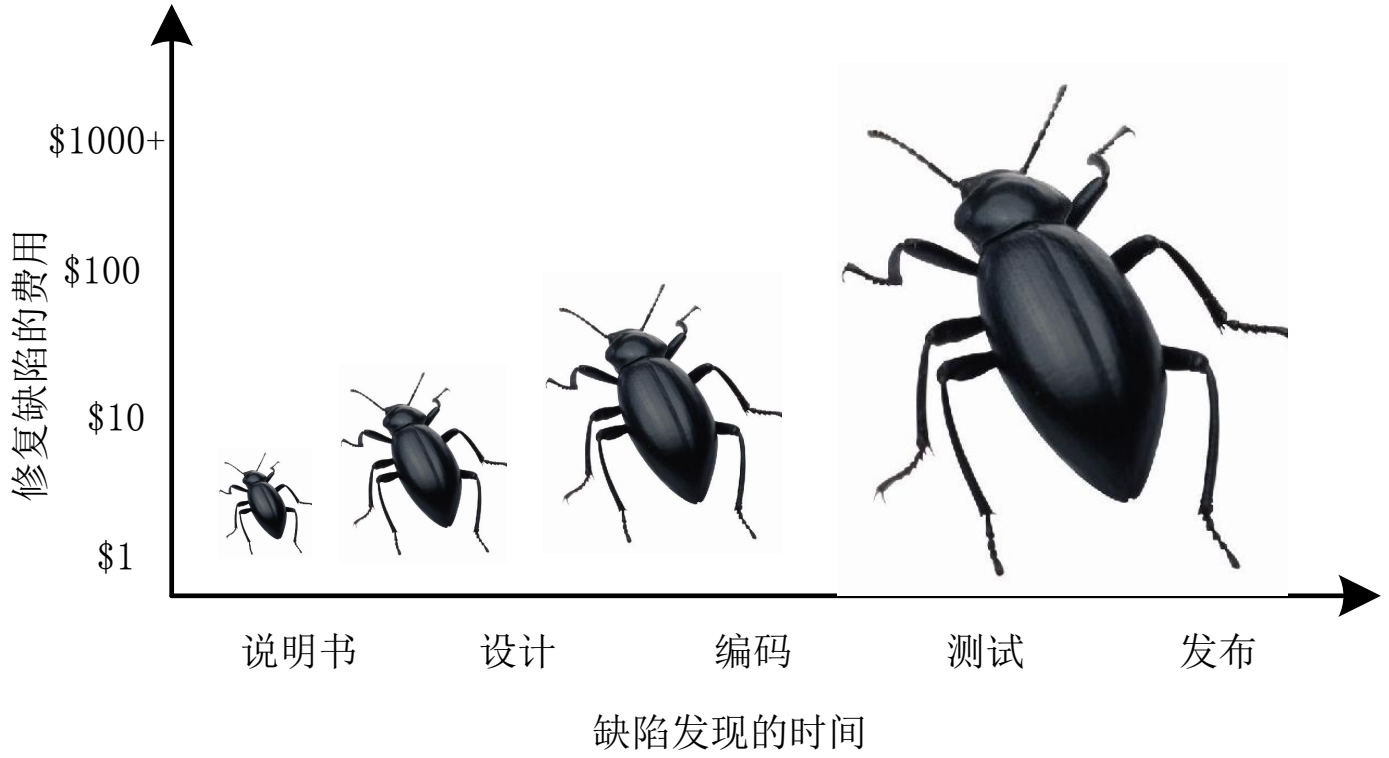
软件测试的意义（续）

软件缺陷的表现形式：

- 软件没有实现需求规格说明所要求的功能模块
- 软件中出现了需求规格说明指明不应该出现的错误
- 软件实现了需求规格说明没有提到的功能模块
- 软件没有实现虽然需求规格说明没有明确提及但应该实现的目标
- 软件难以理解，不容易使用，运行缓慢，或从测试员的角度看，最终用户会认为不好。

高质量的需求规格说明书

软件测试的意义(续)



内容提要

- 1.1 软件测试的意义
- 1.2 软件测试的定义
  - 测试的目的
  - 测试的方法
  - 测试的手段
  - 测试的过程
- 1.3 软件测试的分类
- 1.4 软件测试的过程
- 1.5 小结

1.2 软件测试的定义

- **软件测试 VS 打蚊子**

第1级 初始阶段

•**措施**：测试是完全混乱无序的，测试等同于调试，编码完成后随意地测试和调试，目标是表明软件是奏效的。

•**优势**：最省力气

•**弊端**：开发出的软件产品得不到任何质量的保证，存在很多缺陷，用户无法接受

第1级 守株待兔式

•**措施**：自己静止不动，让蚊子来咬，然后出其不意，将其打死

•**优势**：最省力气

•**弊端**：需要长时间等待；往往存在一击不中，反而白白被蚊子咬的情况

•**结论**：效率不高。灭蚊效果不佳

摘自：武剑洁，吴涛，万琳，《软件测试与质量》，中国大学MOOC，华中科技大学.

软件测试的定义（续）

- **软件测试 VS 打蚊子**

第2级 定义阶段

•**措施**：测试不同于调试。将测试定义为编码完成后的阶段和工作，所有测试都是基于执行的，而且强烈依赖于代码，只有当编码完成后才开始测试，目标是表明软件符合其技术规范。

•**优势**：人们掌握了一定的测试技术和方法，取得了一定的效果

•**弊端**：在需求和设计中没有测试，从而导致大量缺陷扩散到代码中，开发出的软件产品仍然会存在较多缺陷，产品存在质量问题。

第2级 以逸待劳式

•**措施**：点蚊香，将蚊子熏昏过去，让其不能出来咬人

•**优势**：较省力气

•**弊端**：蚊香的味道可能连自己都受不了，而且蚊香会渐渐散去，效果变差，蚊子又飞回来了

•**结论**：灭蚊效果不佳

摘自：武剑洁，吴涛，万琳，《软件测试与质量》，中国大学MOOC，华中科技大学.

软件测试的定义（续）

- **软件测试 VS 打蚊子**

第3级 集成阶段

•措施：将测试集成到整个软件生存周期，开始考虑客户和用户的需求来建立测试目标，将测试看做专业化的活动，成立专门的测试组织，拥有基本的测试工具。。

•优势：基于技术的测试，效果更好。

•弊端：在整个软件生存周期中没有建立正式的评审程序没有开展评审活动，测试组疲于应付

第3级 十八般兵器式

•措施：在房间的不同地方，放不同的武器，例如，徒手打蚊子，用蚊子拍打蚊子，墙角放水盆，床边放灭蚊器，身上涂花露水，挂蚊帐，用用诱捕蚊子的灭蚊器等

•优势：全面撒网，灭蚊效果好

•弊端：蚊子无处不在，打死一只又来一只，无法掌控

•结论：灭蚊效果较好，但疲于应付

摘自：武剑洁，吴涛，万琳，《软件测试与质量》，中国大学MOOC，华中科技大学.

软件测试的定义（续）

- **软件测试 VS 打蚊子**

第4级 管理与测量

•**措施**：测试成为一个可以测量和量化的过程，开发过程引入评审机制，测试用例和测试过程被管理起来。

•**优势**：基于规范的测试，拥有流程控制，出现质量管理活动。

•**弊端**：只能被动地找缺陷，无法主动控制缺陷。

第4级 举一反三式

•**措施**：通过尝试不同的灭蚊方式，不断试验，总结经验，对灭蚊过程及效果进行数据统计，寻找最佳的组合方式

•**优势**：综合多种灭蚊方法的优势，同时降低劳动强度

•**弊端**：只能被动地找蚊子，无法主动控制蚊子的数量

摘自：武剑洁，吴涛，万琳，《软件测试与质量》，中国大学MOOC，华中科技大学.

软件测试的定义（续）

- **软件测试 VS 打蚊子**

第5级 最佳化

•措施： 建立缺陷预防的思想，通过统计抽样等方式不断改进测试，自动工具完全支持测试用例的运行，开展各种与测试有关的度量活动。

•优势： 机制好转，不断改进测试，可以度量和优化产品质量。

第5级 高瞻远瞩式

•措施： 将房间打扫干净，保持室内整洁，少开窗。睡觉前要看看窗户等处是否有蚊子，先行灭之。最后在入睡后使用多种灭蚊方式，保证一夜安睡。

•优势： 不仅注意杀蚊，还有预防的思想，包括防止蚊子进入室内，以及对已经在屋内的蚊子进行预防性查杀。效果最好。

摘自：武剑洁，吴涛，万琳，《软件测试与质量》，中国大学MOOC，华中科技大学.

软件测试的定义(续)

- **IEEE对软件测试的定义：**

  软件测试是使用人工或自动手段来运行或测定某个系统的过程，检验它是否满足规定的需求或者弄清预期结果与实际结果之间的差别。

软件测试的定义(续)

- 软件测试的目的：
  1. 检验软件系统满足需求
  2. 弄清楚预期结果与实际结果之间的差别

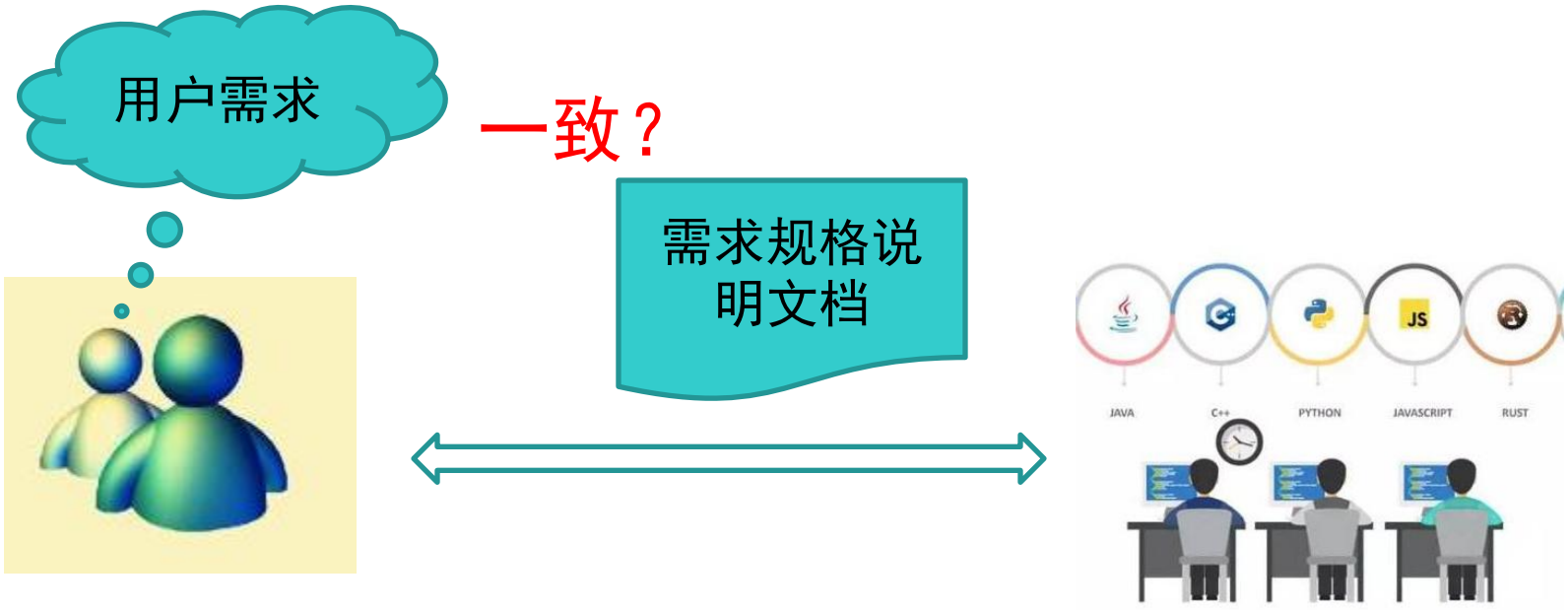
根本目的：确保软件系统满足需求！

软件测试是使用人工或自动手段来运行或测定某个系统的过程，检验它是否满足规定的需求或者弄清预期结果与实际结果之间的差别。

软件测试的定义(续)

- 需求是软件测试的核心

  测试人员如何获取软件系统的需求？



软件测试的定义(续)

- **G. J. Myers（代表作《软件测试的艺术》）对测试的定义：测试是为发现错误而针对某个程序或系统的执行过程**
  - 测试是为了证明程序有错，而不是证明程序无错误
  - 一个好的测试用例可以发现至今尚未发现的错误
  - 一个成功的测试能发现至今未发现的错误。

软件测试的定义(续)

- 软件测试的方法：
  1. 动态测试
  2. 静态测试

软件测试是使用人工或自动手段来运行或测定某个系统的过程，检验它是否满足规定的需求或者弄清预期结果与实际结果之间的差别。

软件测试的定义(续)

- 动态测试

  通过人工或使用工具运行程序，使被测代码在相对真实的环境下运行，观察程序运行时行为，并通过检查、分析程序的执行状态和外部表现，来定位程序的错误

- 设计测试用例
- 运行测试用例
- 校验被测系统的实际输出与预期输出是否一致

软件测试的定义(续)

- 静态测试

  借助专用的软件测试工具评审软件文档或程序，通过分析或检查源程序，发现程序的不足之处

  - 代码检查
  - 静态结构分析
  - 代码质量度量

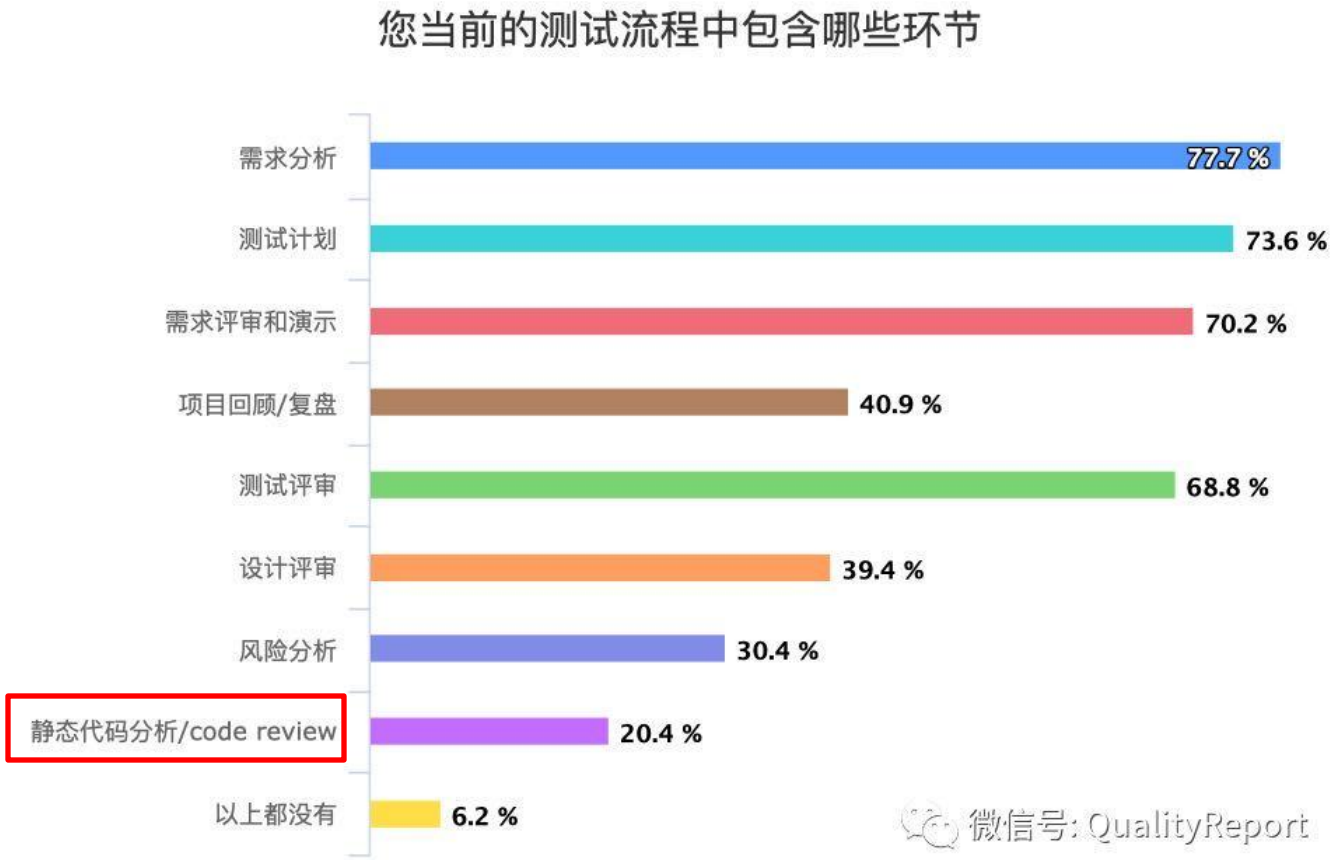
软件测试的定义(续)

动态测试 VS **静态测试**

- 提供被测对象
- 准备相关预期
- 设计测试用例
- 搭建测试环境
- 运行测试用例
- 检查测试结果
- 记录测试过程
- 报告发现的缺陷

- 提供被测对象
- 准备相关预期
- 阅读代码
- 阅读文档
- 报告发现的缺陷

软件测试的定义(续)



2019年软件测试行业调查报告

软件质量报道

软件测试的定义(续)

- **软件测试的手段：**
  1. **人工**
  2. **自动**

软件测试是使用人工或自动手段来运行或测定某个系统的过程，检验它是否满足规定的需求或者弄清预期结果与实际结果之间的差别。

软件测试的定义(续)

手工动态测试

- 提供被测对象
- 准备相关预期
- 设计测试用例
- 搭建测试环境
- 运行测试用例
- 检查测试结果
- 记录测试过程
- 报告发现的缺陷

VS

自动化动态测试

- 提供被测对象
- 准备相关预期
- 设计测试用例
- 搭建测试环境
- 运行测试用例
- 编写测试脚本
- 检查测试结果
- 记录测试过程
- 报告发现的缺陷

软件测试的定义(续)

- 软件测试的流程：
  1. 计划
  2. 设计
  3. 实施
  4. 评估

软件测试是使用人工或自动手段来运行或测定某个系统的过程，检验它是否满足规定的需求或者弄清预期结果与实际结果之间的差别。

软件测试的定义(续)

- 计划->设计 ->实施->评估
  - 谁
  - 何时
  - 使用什么方法
  - 用到什么资源
  - 遵循什么标准
  - 对什么展开测试
  - 可能有哪些风险

软件测试的定义(续)

- 计划->设计 ->实施->评估
  - 设计测试用例
  - 设计测试过程

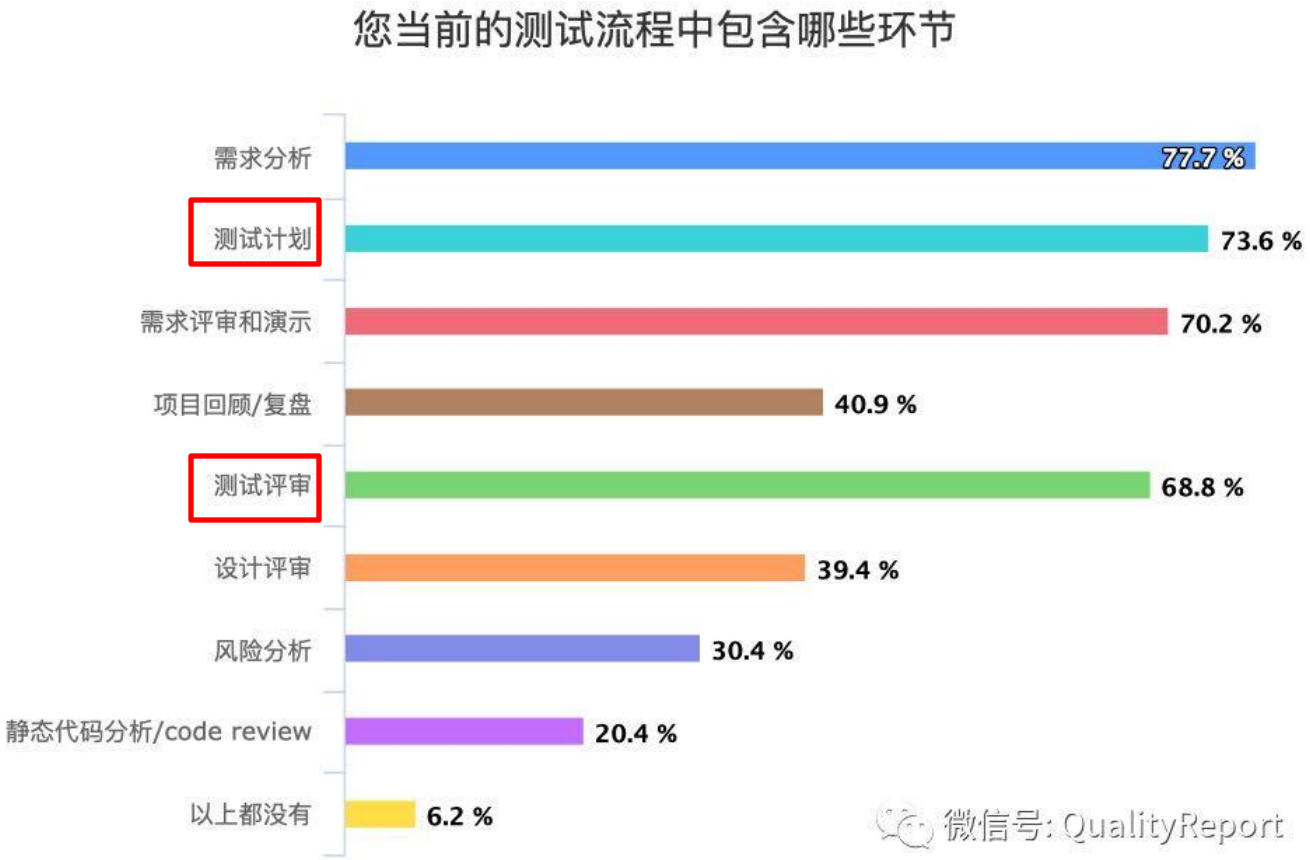
软件测试的定义(续)

- 计划->设计 ->实施->评估
  - 运行测试用例
  - 检查测试结果
  - 提交缺陷报告

软件测试的定义(续)

- 计划->设计 ->实施->评估
  - 评估测试工作
  - 评估被测系统

软件测试的定义(续)



2019年软件测试行业调查报告

软件质量报道

软件测试的定义(续)

- 我们对软件做测试的根本目的是什么？

A. 提高软件的质量
B. 确保程序符合用户的需求
C. 找出程序中的缺陷
D. 确保程序没有缺陷

软件测试的定义(续)

- 从软件测试的定义理解测试：
- 根本目的：验证需求
- 核心关键：测试设计
- 整体思路：比较预期输出与实际输出
- 存在不足：没有说明具体如何做测试

软件测试的定义(续)

- 从软件测试的定义+缺陷的表现理解测试：
- 细化整体思路：体现如何比较预期输出与实际输出
- 问题：测试就是以需求规格说明书为依据，穷尽检查每一项需求？

软件测试的定义(续)

- **软件测试的理想：**

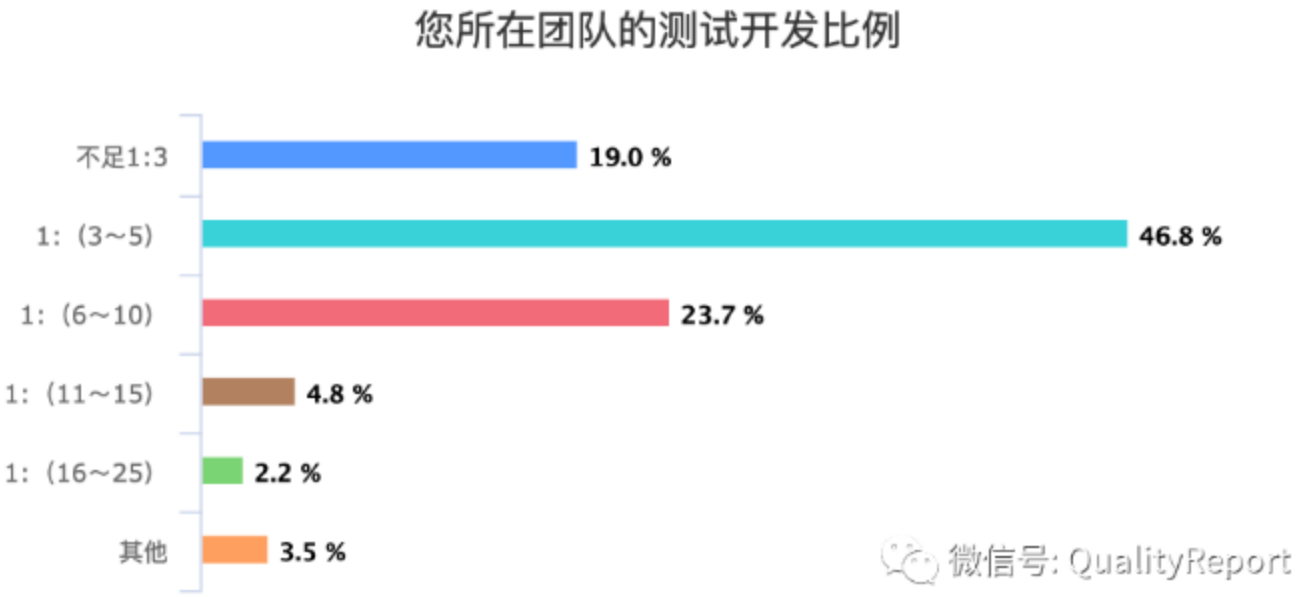
  在充足的时间内，有足够的成本完成测试，保证软件具备良好的质量。

软件测试的定义(续)

- 软件测试的现实：
  - 时间

软件测试的定义(续)

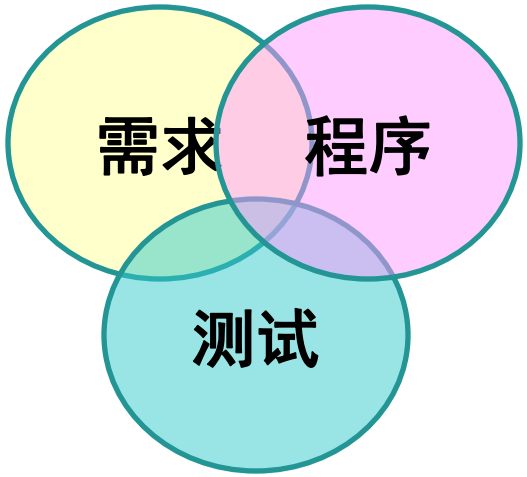
- 软件测试的现实：
  - 成本控制



2019年软件测试行业调查报告
软件质量报道

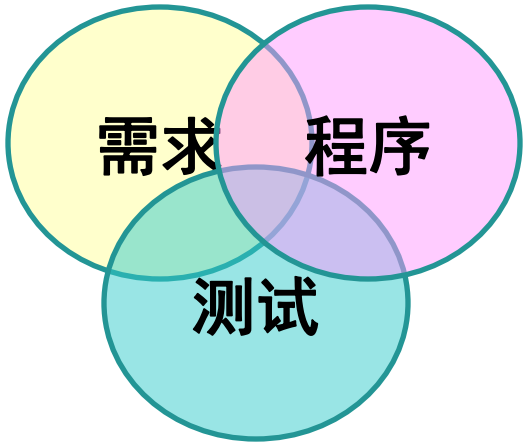
软件测试的定义(续)

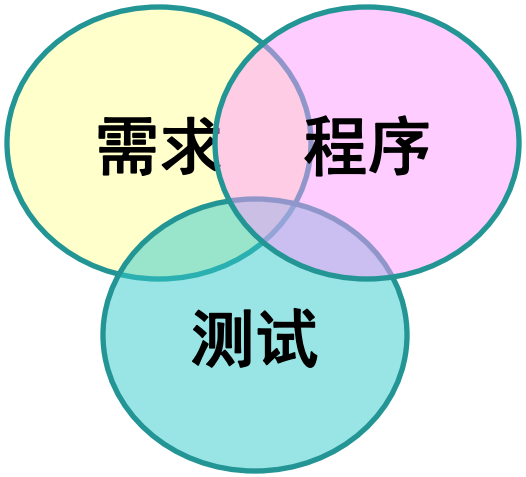
- 软件测试的现实：
  - 软件质量要求



软件测试的定义(续)

- 软件测试的现实：
  - 软件质量要求





软件测试的定义(续)

测试可以证明软件无错吗？

E.W. Dijkstra:

“测试可以表明缺陷的存在，但绝不能证明没有缺陷”

软件测试的定义(续)

- 软件测试的现实：
- 需求无法穷尽
- 程序无法穷尽
- 缺陷数量未知
- 时间、成本有限

软件测试的定义(续)

- 软件测试的目标：

  在最短时间内，找到最严重、最多的缺陷，最大程度地保证产品符合已知用户需求。

测试用例的设计

软件测试的定义(续)

- 测试用例：

  为了特定测试目的（如考察特定程序路径或验证某个产品特性）而设计的测试条件、测试数据及与之相关的操作过程、期望结果。

软件测试的定义(续)

- 测试用例的基本属性
  - 典型性
  - 可测试性
  - 可重现性

软件测试的定义(续)

测试用例的设计

- 输入数据
  - 正常数据
  - 错误数据
    - 满足数据类型，不在有效取值范围内
    - 不完全满足数据类型
    - 输入条件缺失
  - 边界数据

软件测试的定义(续)

一个简单的测试用例

【测试用例 1】

测试目标：验证输入错误的密码是否有正确的响应。

测试环境：Windows XP 操作系统和浏览器 Firefox 3.0.3

输入数据：用户邮件地址和口令

步骤：

1. 打开浏览器
2. 点击页面右上角的“登录”链接，出现登录界面。
3. 在电子邮件的输入框中输入：test@gmail.com
4. 在口令后面输入：xxxabc
5. 点击“登录”按钮

期望结果：

登录失败，页面重新回到登录页面，并提示“用户密码错误”。

内容提要

- 1.1 软件测试的意义
- 1.2 软件测试的定义
- 1.3 软件测试的分类
- 1.4 软件测试的过程
- 1.5 小结

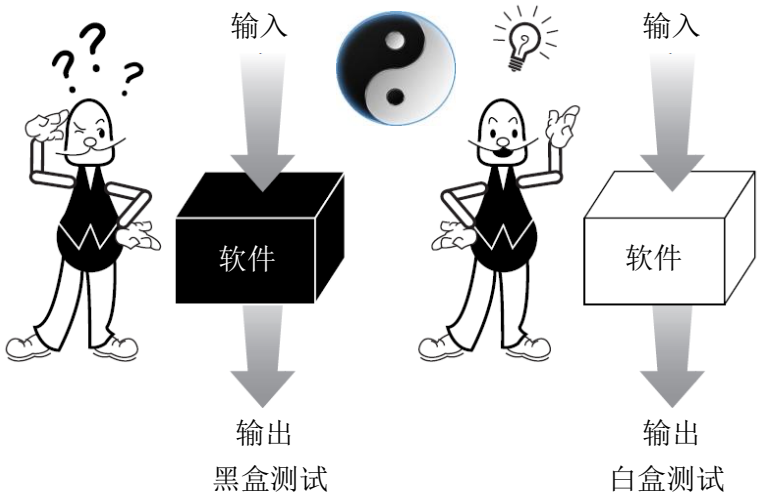
软件测试的分类

- 从测试技术的角度
- 从测试目标的角度
- 从测试执行方式的角度
- 从测试阶段的角度

软件测试的分类（续）

从测试技术的角度：

- 黑盒测试
  关注的是与产品的外部行为相关的缺陷，不考虑内部逻辑结构。
- 白盒测试
  关注的是与代码内部结构相关的缺陷，需要测试人员掌握一定的编程技术。
- 灰盒测试
  关注输出对于输入的正确性，也关注程序的内部表现。



软件测试的分类（续）

从测试目标的角度：

- 功能测试：验证每个功能是否按照事先定义的要求正常工作
- 性能测试：测定系统在不同负载条件下的具体性能指标
- 压力测试：检查系统在高负载、极限负载条件下的系统运行情况
- 安全性测试：测试系统权限设置有效性、应对非授权的内部/外部访问、防范非法入侵、数据备份等

软件测试的分类（续）

从测试目标的角度：

- 可靠性测试：测试系统能否保持长期稳定、正常的运行
- 可恢复性测试：测试系统崩溃、硬件故障等灾难发生后重新恢复系统和数据的能力
- 配置测试：测试系统在不同运行环境下能否正确工作
- 安装测试：在目标环境中测试系统的安装过程
- 回归测试：软件版本修改后的重新测试

软件测试的分类（续）

从测试执行方式的角度：

- 手动测试
- 自动化测试
- 半自动化测试

软件测试的分类（续）

从测试阶段的角度：

- 单元测试
- 集成测试
- 系统测试
- 验收测试

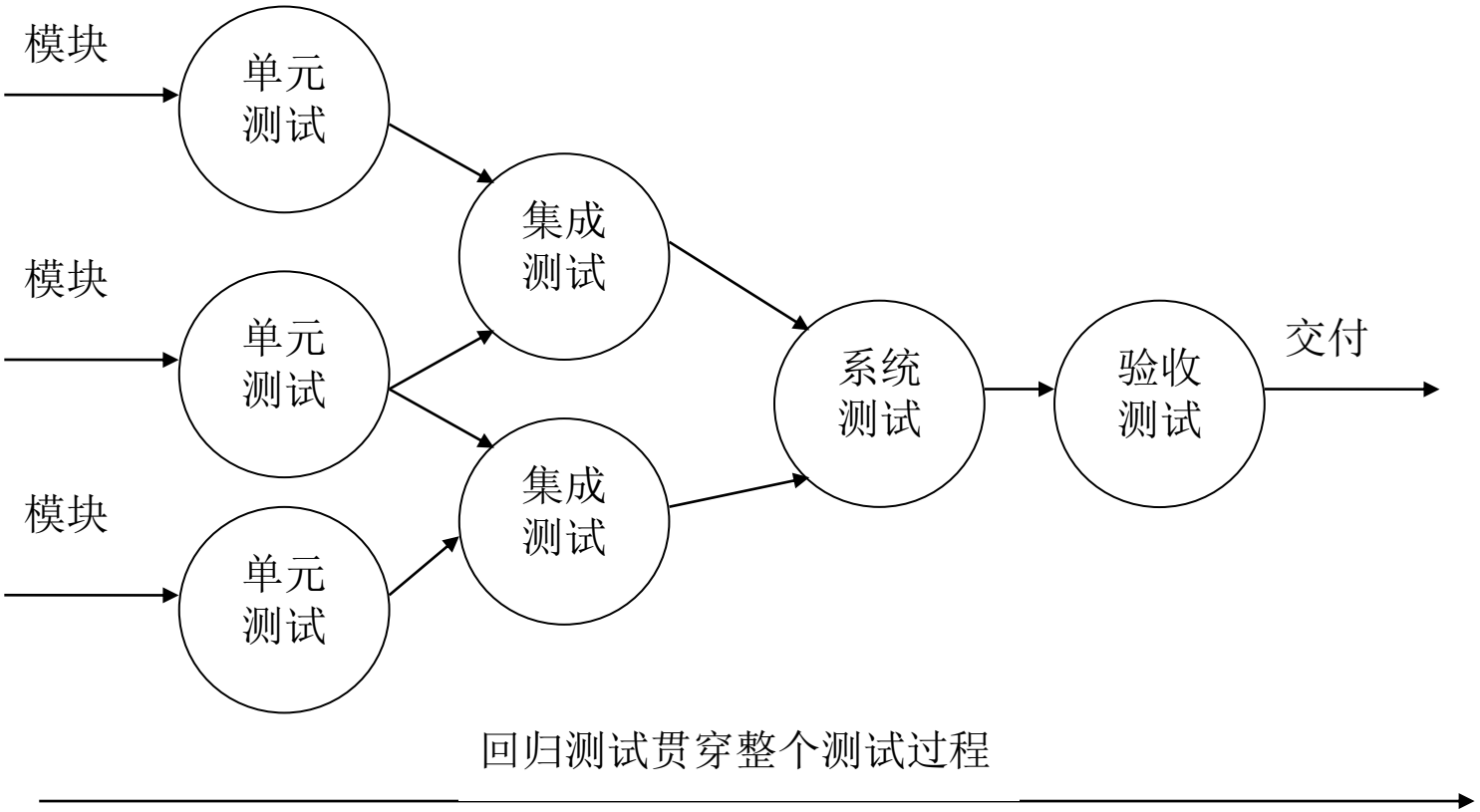
内容提要

- 1.1 软件测试的意义
- 1.2 软件测试的定义
- 1.3 软件测试的分类
- 1.4 软件测试的过程
- 1.5 小结

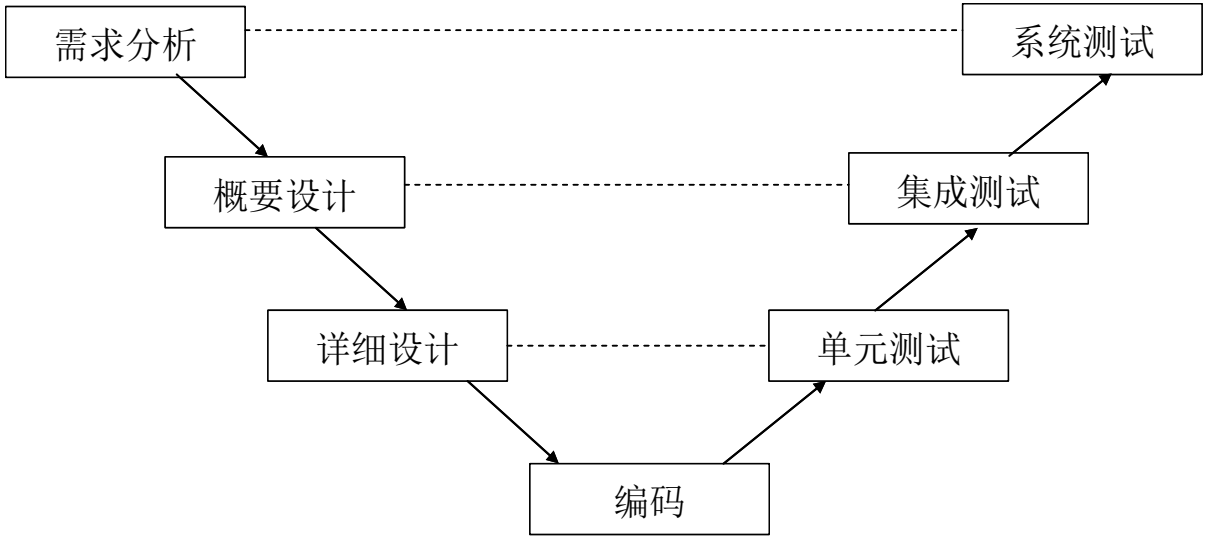
1.4 软件测试的过程

- **软件测试时需要以下三类信息：**
  - 软件配置：指需求说明书、设计说明书和源程序等。
  - 测试配置：指测试方案、测试用例和测试驱动程序等。
  - 测试工具：指计算机辅助测试的有关工具。
- **软件测试是一个综合测试的过程**

软件测试过程(续)



软件测试过程(续)



软件测试过程V模型

软件测试过程(续)

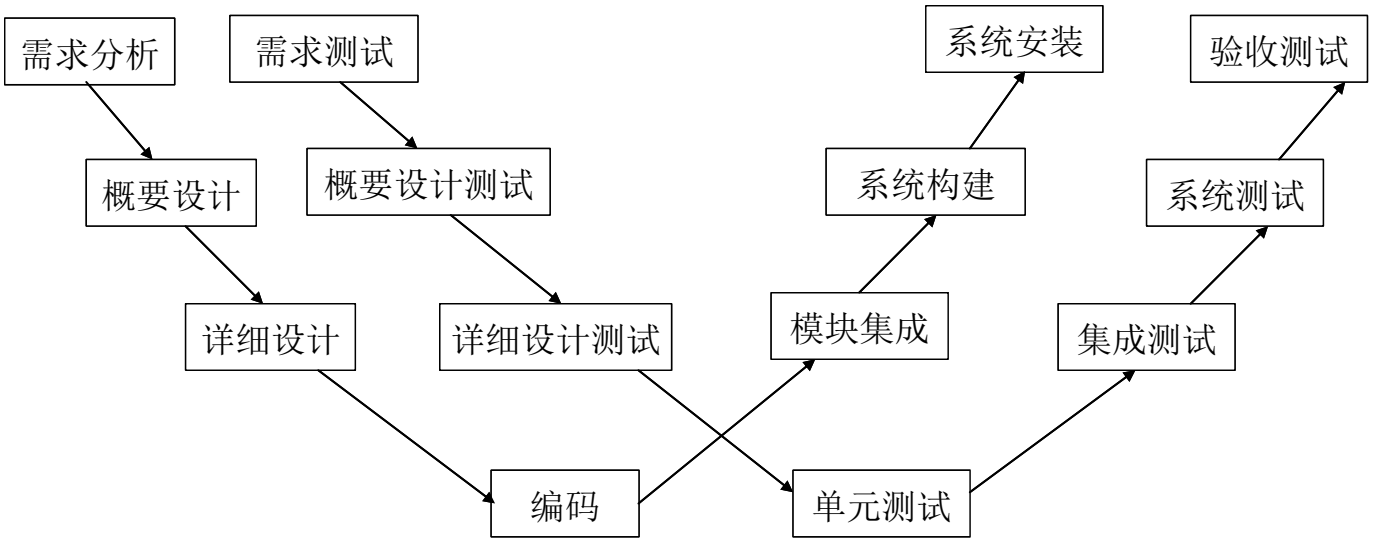


图1.4 软件测试过程W模型

软件测试过程(续)

- **软件测试结果**
  - 总是出现需要修改设计的严重错误
  - 软件功能能正确完成，出现的错误易修改
  - 测试发现不了错误

单元测试

- 定义：对软件中的最小可测试单元或基本组成单元进行检查和验证
- 单元测试与软件开发过程中的详细设计阶段相对应
- 如何定义单元？
  - 具有明确的功能定义
  - 具有明确的性能定义
  - 具有连接其他部门的接口定义
  - 可以清晰地与其他单元区分开来

单元测试(续)

单元选取原则

- **对面向过程的开发语言来说**
  - 单元常指一个函数或子过程
  - 特殊情况：若几个函数之间具有强耦合性，应将它们共同作为一个单元
- **对面向对象的开发语言来说**
  - 单元一般指一个类
  - 特殊情况：若某些基础类非常庞大，此时的测试应上升到集成测试的层面
- **对图形化软件，单元常指一个窗体或一个菜单**

单元测试(续)

测试内容

- 接口测试
- 局部数据结构测试
- 重要执行路径测试
- 错误处理测试
- 边界条件测试

单元测试(续)

- **接口测试：对通过被测模块的数据流进行测试，以检查数据能否正确地输入和输出**
  - 输入的实参与形参是否匹配
  - 被测模块调用其他模块时，传递的实参与被调用模块的形参是否匹配
  - 是否存在于当前入口点无关的参数引用
  - 是否修改了只做输入用的只读形参
  - 全局变量在每个模块中的定义是否一致
  - 是否将某些约束条件作为形参来传递

单元测试(续)

- **局部数据结构测试：检查局部数据结构以保证临时存储在模块内的数据在代码执行过程中是完整和正确的**
  - 是否存在不正确、不一致的数据类型说明
  - 是否存在不正确的变量名
  - 是否存在未初始化或未赋值的变量
  - 是否出现上溢、下溢或地址异常

单元测试(续)

- **重要执行路径测试：最常用和最有效的测试技术，以发现因错误的计算、错误的比较和不适当的控制流而导致的缺陷**
  - 是否存在错误的逻辑运算符或优先次序
  - 是否存在被零除的风险
  - 运算精度是否足够
  - 变量的初值是否正确
  - 表达式的符号是否正确
  - 是否存在不同数据类型变量之间的比较
  - 是否存在因运算精度不够，致使期望值与实际值不相等的两值比较
  - 关系表达式中是否存在错误的变量和比较符
  - 是否存在不可能的循环终止条件
  - ……

单元测试(续)

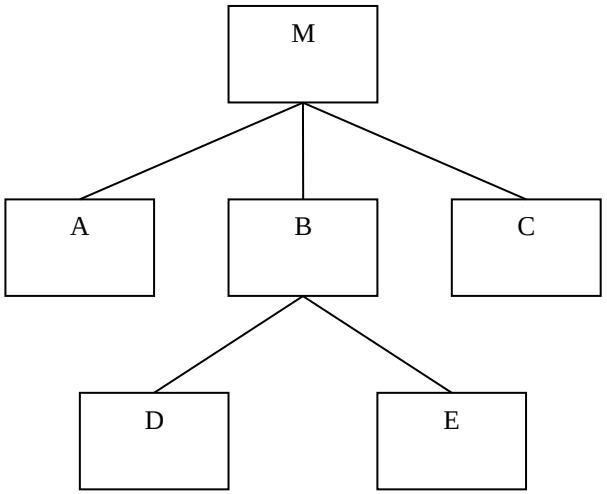
- **错误处理测试：测试程序处理错误的能力**
  - 输出的出错信息是否难以理解，提供的信息是否充足
  - 显示的错误是否与实际遇到的缺陷符合
  - 对错误的处理是否正确
  - 在程序自定义的出错处理运行之前，缺陷条件是否已经引起系统干预

单元测试(续)

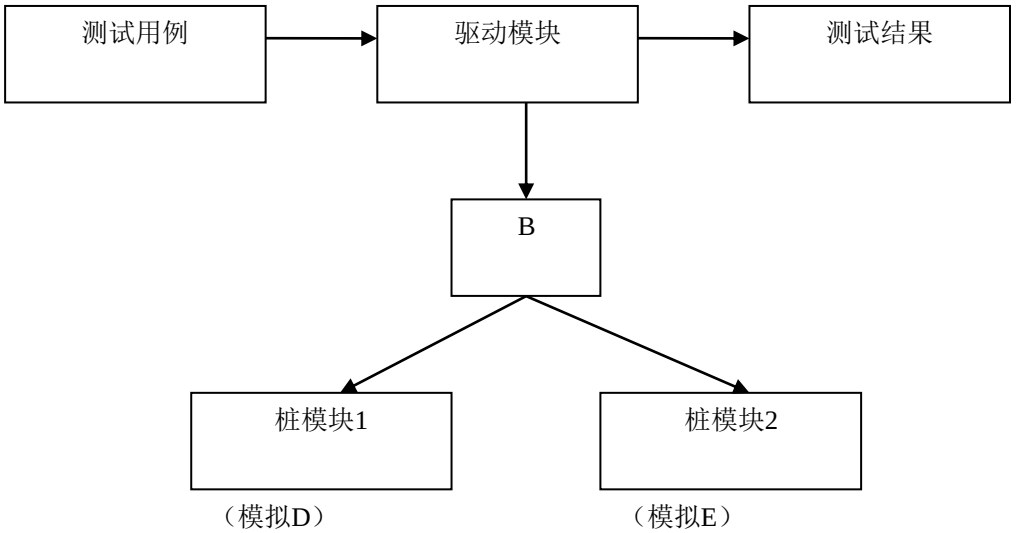
- **边界条件测试：程序最容易在边界上出错**
  - 输入/输出数据的等价类边界
  - 选择条件和循环条件的边界
  - 复杂数据结构的边界

单元测试(续)

- **测试方法：在对模块进行测试时，它不能独立运行，需要设计辅助测试模块**
  - 驱动模块(Driver)：模拟被测模块的上级调用模块
  - 桩模块(Stub)：模拟被测模块在执行过程中所要调用的模块



（a）软件结构



（b）模块B的测试环境

单元测试(续)

- **测试技术**
  - 静态测试
  - 白盒测试
  - 状态转换测试：模拟使状态发生转换的各种用户操作场景，以及通过一些非正常手段来校验不允许发生的状态转换
  - 功能测试和非功能测试

单元测试(续)

- **测试人员**
  - 单元的开发人员设计测试用例、执行测试、修改缺陷
  - 开发组长监督
  - 用户代表观察

集成测试

- 定义：在单元测试的基础上，将所有已通过单元测试的模块按照概要设计的要求组装为子系统或系统，确保各单元模块组合在一起后能够按既定意图协作运行，并确保增量的行为正确

  未经单元测试的模块不应进行集成测试

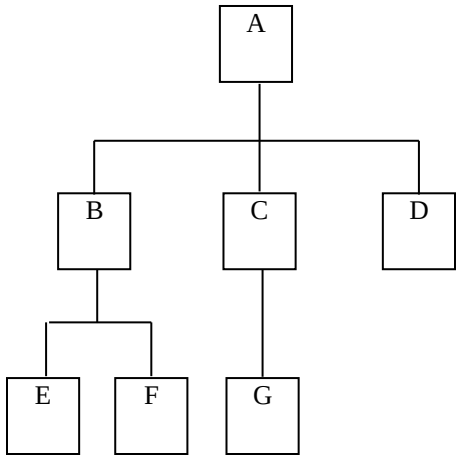
- 集成测试与软件开发过程中的概要设计阶段相对应

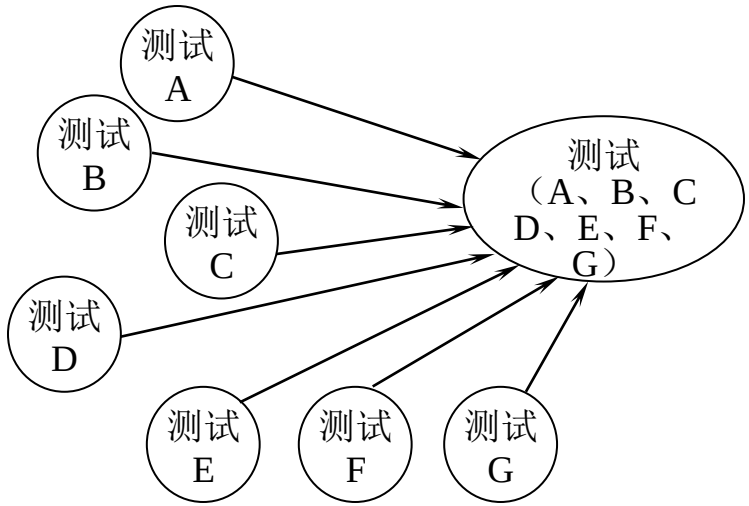
集成测试(续)

- **测试内容：包括模块之间的接口以及集成后的功能**
  - 将各模块连接起来时，穿越模块接口的数据是否会丢失
  - 各子功能组合起来能否达到预期要求的父功能
  - 一个模块的功能是否会对其他模块的功能产生不利影响
  - 全局数据结构是否有问题，是否会被异常修改
  - 单个模块的误差累积起来，是否会放大到不可接受的程度

集成测试(续)

- 测试方法
  - **非增量式集成测试方法**
    - 首先将各模块独立地进行单元测试，然后把所有模块组装在一起进行测试
    - 容易出现混乱





集成测试(续)

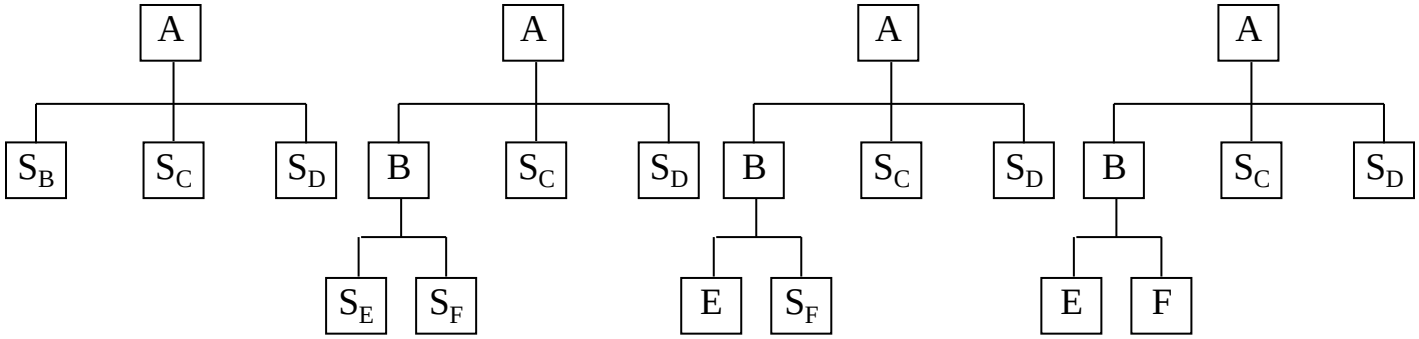
- 测试方法
  - **增量式集成测试方法**：首先。。。然后将各模块逐步组装成较大的系统，在组装的过程中边组装边测试
    - 自顶向下集成
    - 自底向上集成
    - 三明治集成

集成测试(续)

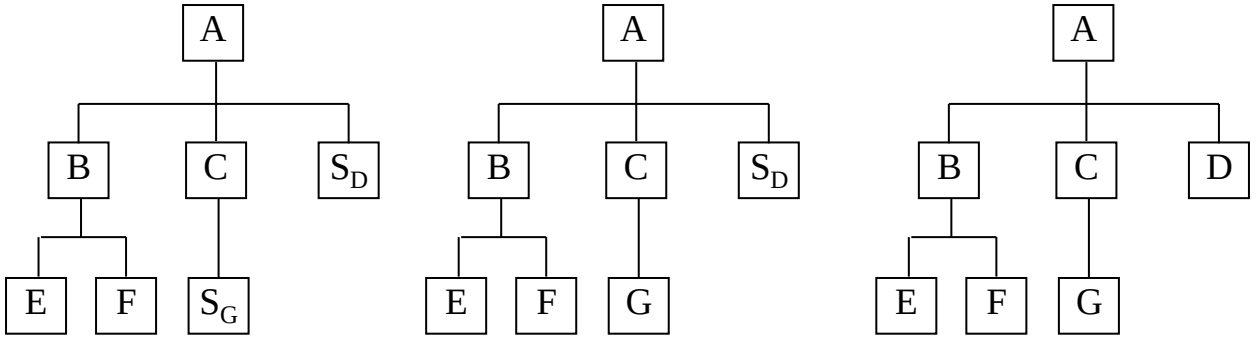
- **自顶向下增量式集成测试**
  - **测试方法**
    - 首先利用桩模块测试主模块，通过测试后，用实际的模块替代桩模块进行测试
    - 重复上面步骤，直至替代了所有桩模块
  - **决定模块测试次序的基本原则**
    - 尽早测试关键的模块
    - 尽早测试包含输入、输出功能的模块
    - 深度优先
    - 宽度优先

集成测试(续)

- 自顶向下增量式集成测试



（a）测试A　（b）测试AB　（c）测试ABE　（d）测试ABEF



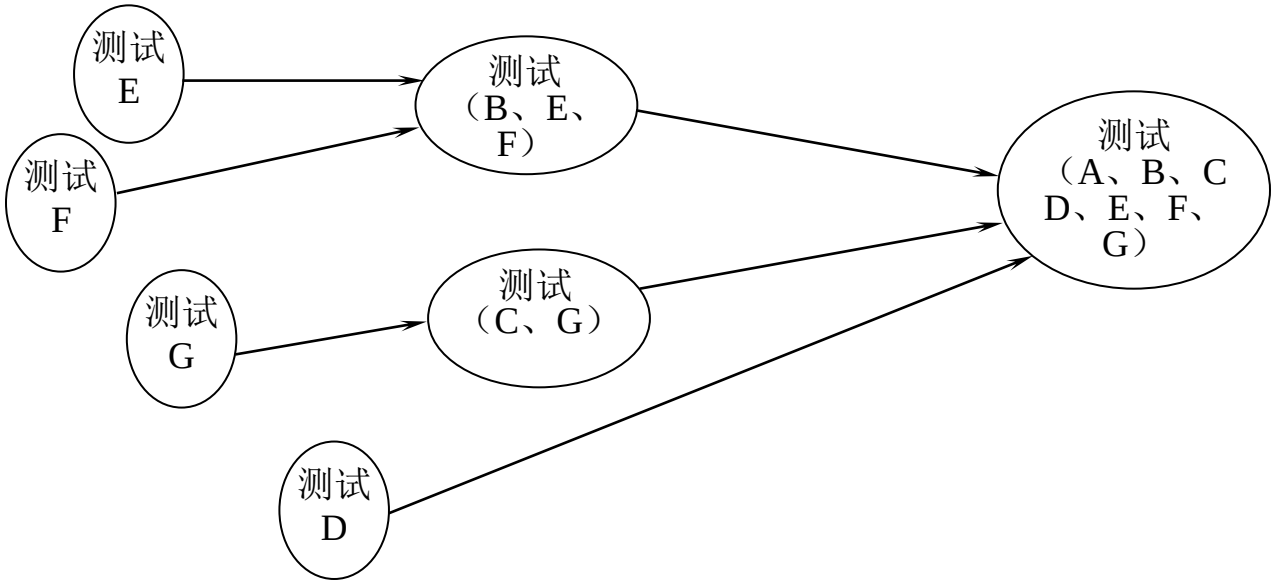
（e）测试ABEFC　（f）测试ABEFCG　（g）测试ABEFCGD

集成测试(续)

- **自底向上增量式集成测试**
  - **测试方法**
    - 首先利用驱动模块测试最底层模块，通过测试后，用实际的模块替代驱动模块进行测试
    - 重复上面步骤，直至替代了所有驱动模块
  - **决定模块测试次序的基本原则**
    - 该模块的所有下级模块都已测试过了

集成测试(续)

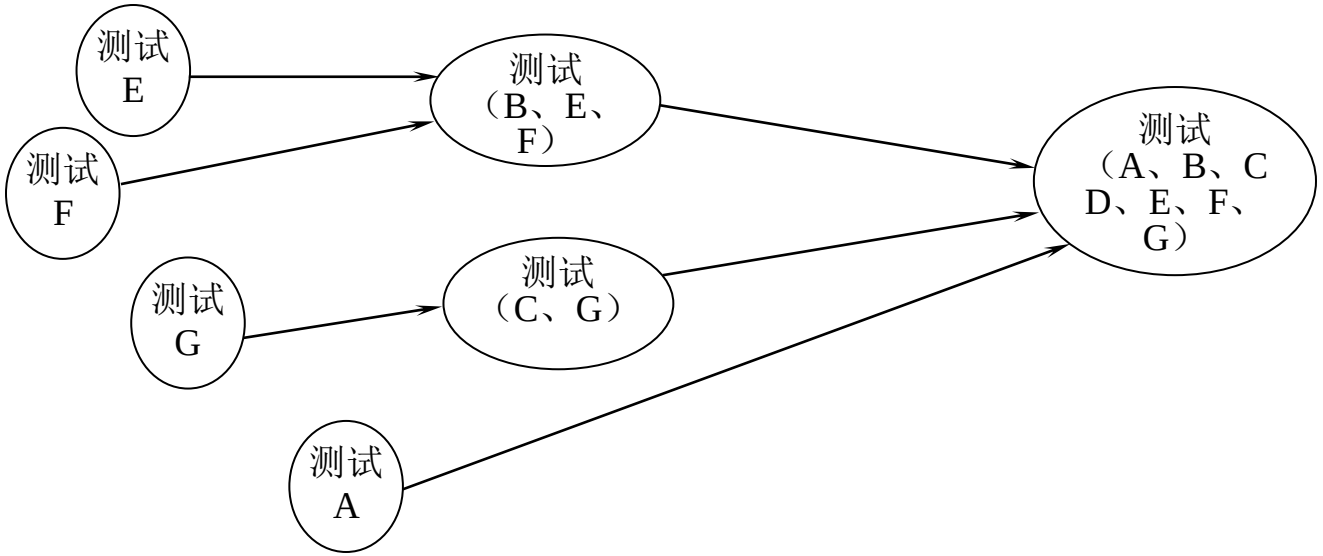
- 自底向上增量式集成测试



集成测试(续)

- 三明治集成测试

  以B模块所在层为界



集成测试(续)

测试方法	优点	缺点
自顶向下	1. 如果程序错误趋向于发生在程序的顶端时，有利于查出错误。 2. 可以较早出现程序的轮廓。 3. 加进输人 / 输出模块后，较方便描述测试用例。	1. 桩模块较难设计。 2. 模块介入使结果较难观察。
自底向上	1. 如果程序错误趋向于发生在程序的底端时，有利于查出错误。 2. 容易产生测试条件和观察测试结果 3. 容易编写驱动模块。	1. 在加入最后一个模块之前，程序不能作为一个整体存在。 2. 必须给出驱动程序。

集成测试(续)

- **三明治集成测试**
  - 一种混合增量式测试策略，综合了自顶向下和自底向上两种集成方法
  - 这种方法桩模块和驱动模块的开发工作都比较小，不过代价是在一定程度上增加了定位缺陷的难度。

集成测试(续)

- **测试技术：多为黑盒测试，适当辅以白盒测试**
  - 确认组成一个完整系统的模块之间的关系
  - 评审模块之间的交互和通信需求，确认模块间的接口
  - 使用上述信息产生一套测试用例
  - 采用增量式测试

集成测试(续)

- 测试人员
  - 开发人员执行
  - 开发组长监督
  - 独立测试观察员监控
  - 用户代表观察

系统测试

- 定义：将已经过良好的集成测试的软件系统，作为整个计算机系统的一部分，在实际使用环境下进行一系列的严格测试
- 与集成测试、单元测试的不同
  - 在实际的用户使用环境下执行
  - 涉及软件、硬件等多方面因素

系统测试(续)

- **测试技术**

  完全采用黑盒测试技术

- **测试人员**
  - 独立的测试小组执行
  - 测试组长监督
  - 独立测试观察员监控
  - 用户代表观察

验收测试

- 定义：以用户为主，软件开发人员、实施人员和质量保证人员共同参与的有效性测试或合格性测试，让用户决定是否接受产品
- 测试内容
  - 软件系统
  - 文档

验收测试(续)

- **测试过程**
  - 明确规定验收测试通过的标准
  - 确定验收测试方法
  - 确定验收测试的组织和可利用的资源
  - 确定测试结果的分析方法
  - 制定验收测试计划并进行评审
  - 设计验收测试的测试用例
  - 审查验收测试的准备工作
  - 执行验收测试
  - 分析测试结果，决定是否通过验收

验收测试(续)

- **测试技术**
  - 完全采用黑盒测试
  - α测试：软件开发公司内模拟软件系统运行环境的一种验收测试，测试后调整的软件产品成为β版本
  - β测试：软件开发公司组织各方面的典型用户在日常工作中实际使用β版本，并要求用户报告异常情况
- **测试人员**
  - 测试组协助
  - 用户代表执行
  - 测试组长负责

回归测试

- 定义：修改旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误
- 测试策略
  - 测试用例库的维护
  - 回归测试包的选择

回归测试(续)

- **测试用例库的维护**
  - 删除过时的测试用例
  - 改进不受控制的测试用例
  - 删除冗余的测试用例
  - 增添新的测试用例

回归测试(续)

- **回归测试包的选择**
  - 再测试全部用例
  - 基于风险选择测试
  - 基于操作剖面选择测试
  - 再测试修改的部分

回归测试(续)

- **测试过程**
  - 识别出软件中被修改的部分
  - 从原基线测试用例库T中，排除所有不再适用的测试用例，建立一个新的基线测试用例库T’
  - 依据一定的策略，从T’中选择测试用例集T0来测试被修改的软件
  - 如果必要，生成新的测试用例集T1，用于测试T0无法重复测试的软件部分
  - 用T0 ∪ T1测试修改后的软件

回归测试(续)

- **测试技术**
  - 黑盒测试
  - 白盒测试
  - 非功能测试
  - 错误猜测
- **测试人员**
  - 几乎所有的软件开发人员参与
  - 开发组长负责
  - 独立测试观察员观察

软件测试的原则

1. 在整个开发过程中要尽早地和不断地进行软件测试。
2. 在开始测试时，不应默认程序中不存在错误。
3. 设计测试用例时，要给出测试的预期结果。
4. 测试工作应避免由系统开发人员或开发机构本身来承担。
5. 对合理的和不合理的输入数据都要进行测试。

软件测试的原则(续)

6. 重点测试错误群集的程序区段。
7. 除检查程序功能是否完备外，还要检查程序功能是否有多余。
8. 用穷举测试是不可能的。
9. 长期完整保留所有的测试用例和测试文件，直至该软件产品被废弃为止。

内容提要

- 1.1 软件测试的意义
- 1.2 软件测试的定义
- 1.3 软件测试的分类
- 1.4 软件测试的过程
- 1.5 小结

1.5 小结

- 软件缺陷
- 软件测试的定义
- 软件测试的分类
- 软件测试的过程