



软件测试

张程

Email: bootan@cqu.edu.cn

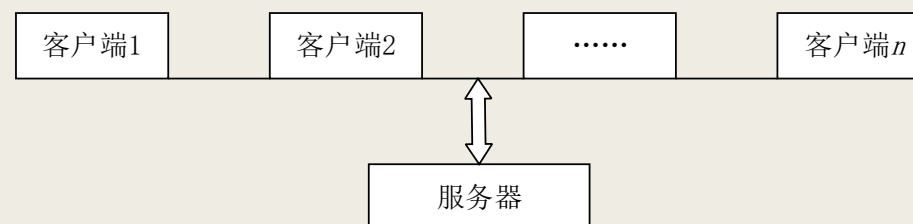
QQ:80463125

主流信息应用系统测试

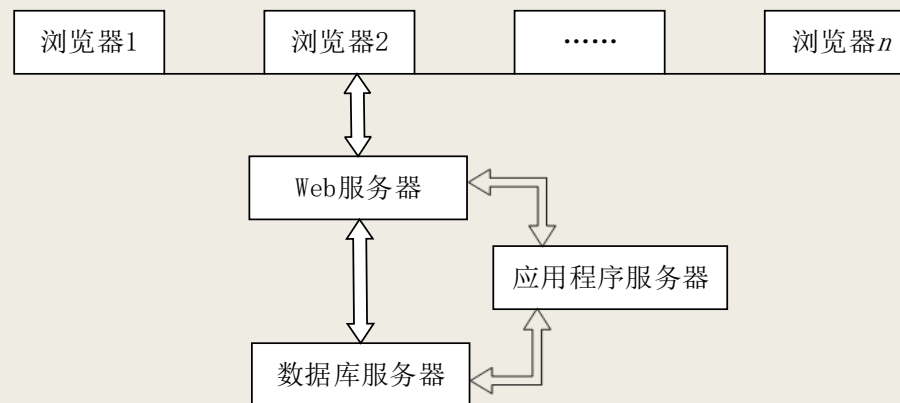
Web应用系统测试

■ Web系统基本组成

- C/S结构(Client/Server, 客户端/服务器)



- B/S结构(Browser/Server, 浏览器/服务器)



Web应用系统测试

- 一个典型的Web系统包括
 - 访问客户端
 - Web应用服务器
 - 数据库
 - 网络及中间件
 - 防火墙与CA认证
 - 一些大型Web系统为了承受较大的访问压力，会采用负载均衡技术，使用多个Web应用服务器分担来自客户端的访问压力。

Web应用系统测试-综述

■ Web系统的服务器端应用特点

- 不存在与系统最终用户相交互的用户界面
 - 客户端通过通信协议、应用编程接口和其他接口标准与服务器端应用进行交互以调用其功能和访问数据
- 自动运行
 - 对于测试人员来说，服务器端应用就是一个黑盒子
 - 一种用来提高错误重现能力的方法是记录事件日志
 - 应用日志允许跟踪由具体应用生成的事件

■ Web应用采用基于组件的体系结构，容易出现错误共享，这种问题对测试有两个主要的影响

- 已有对象或组件在被其他的应用或对象引用时必须经过了全面测试
- 必须进行过全面的回归测试

■ 标准的Web应用系统通常使用关系型数据库，对数据的访问和操纵比使用文件系统更高效

■ Web系统的互操作性

- 系统或系统中的组件与其他系统或组件相交互和无缝协作的能力
- 可能导致在组件之间的通信出现信息丢失或误解的情况



Web应用系统测试

■ Web系统质量测试分类

- 系统架构
 - 客户端的测试
 - 服务器端的测试
 - 网络的测试
- 职能
 - 应用功能的测试
 - Web应用服务的测试
 - 安全系统的测试
 - 数据库服务的测试

■ 软件的质量特性

- 功能测试
- 性能测试
- 安全性测试
- 兼容性测试
- 易用性测试

Web应用系统测试的实施

■ 功能测试

- 链接测试

链接是Web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段

- 测试所有链接是否按指示的那样确实链接到了该链接的页面
- 测试所链接的页面是否存在
- 保证Web应用系统上没有孤立的页面

- 表单测试

用户给Web应用系统管理员提交信息时，就需要使用表单操作

- 测试提交操作的完整性，以校验提交给服务器的信息的正确性
- 检验默认值的正确性

-

Web应用系统测试的实施-功能测试

- Cookie 测试
 - Cookie通常用来存储用户信息和用户在某应用系统的操作，当一个用户使用Cookie访问了某一个应用系统时，Web服务器将发送关于用户的信息
 - 把该信息以Cookie的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登录等信息
 - 测试Cookie是否起作用
 - 测试Cookie是否按预定的时间进行保存
 - 刷新对Cookie有什么影响
- 应用程序特定的功能需求
 - 除了以上基本的功能测试外，测试人员仍需要对应用程序特定的功能需求进行验证。



Web应用系统测试的实施

■ 性能测试

- 连接速度测试

用户连接到Web应用系统的速度根据上网方式的变化而变化

- Web系统响应时间太长，用户将离开页面
- 页面如果有超时限制，用户将重新登录
- 连接速度太慢，可能引起数据丢失，得不到真实的页面

- 负载测试

测量Web系统在某一负载级别上的性能，以保证Web系统在需求范围内能正常工作

- 负载级别可以是某个时刻同时访问Web系统的用户数量，也可以是在线数据处理的数量
- 负载测试应该安排在Web系统发布以后，在实际的网络环境中进行测试

- 压力测试

指实际破坏一个Web应用系统，测试系统的反映

- 测试系统的限制和故障恢复能力
- 测试的区域包括表单、登录和其他信息传输页面等

Web应用系统测试的实施

■ 可用性测试

- 导航测试：导航描述了用户在一个页面内操作的方式，在不同的用户接口控制之间或在不同的连接页面之间
 - Web应用系统导航帮助要尽可能地准确
 - 确保用户凭直觉就知道Web应用系统里面是否还有内容，内容在什么地方
- 图形测试：适当的图片和动画既能起到广告宣传的作用，又能起到美化页面的功能。
- 内容测试：检验Web应用系统提供信息的正确性、准确性和相关性
 - 信息的正确性是指信息是可靠的还是误传的
 - 信息的准确性是指是否有语法或拼写错误
 - 信息的相关性是指是否在当前页面可以找到与当前浏览信息相关的信息列表或入口
- 表格测试：验证表格是否设置正确
- 整体界面测试：整个Web应用系统的页面结构设计，是给用户的一个整体感，其实是一个对最终用户进行调查的过程



Web应用系统测试的实施

■ 客户端兼容性测试

- 平台测试：最常见的操作系统有Windows、iOS、Linux
 - 在各种操作系统下对Web系统进行兼容性测试
 - 浏览器测试：浏览器是Web客户端最核心的构件
 - 来自不同厂商的浏览器对Java、JavaScript、Activex、plug-ins或不同的HTML规格有不同的支持
 - 框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不显示
 - 不同的浏览器对安全性和Java的设置也不一样
 - 分辨率测试
 - 页面版式在 640×400 、 600×800 或 1024×768 的分辨率模式下是否显示正常
 - 字体是否太大或太小以至于无法浏览
 - 文本和图片是否对齐
 - 打印测试
 - 屏幕上显示的图片 and 文本的对齐方式可能与打印出来的东西不一样
 - 组合测试
- 例如： 600×800 的分辨率在MAC机上可能不错，但是在IBM兼容机上却很难看



Web应用系统测试的实施

- 安全性测试
 - 注册登录测试
 - 超时限制测试
 - 日志文件测试
 - 加密测试
 - 授权测试
- 接口测试
 - 服务器接口
 - 外部接口
 - 错误处理
- 故障恢复测试
 - 确保系统能从各种意外数据损失或完整性破坏的各种软/硬件故障中恢复
 - 核实系统能够是否能在客户/服务器断电、网络通信中断、异常关闭某个功能、错误的操作顺序这4种状况下正确恢复到预期的已知状态



数据库测试

■ 数据库测试概述

- 数据库技术的广泛使用直接导致了联机分析处理、数据仓库和数据挖掘等技术的出现，促使数据库向智能化方向发展。
- 越来越多的数据库操作被作为存储过程直接放在数据库上执行以提高执行效率和提高安全性，或者数据库的相关操作作为独立的逻辑层存在。
- 随着数据库开发的日益升温和数据库系统的复杂化，数据库测试也需要独立出来进行符合数据库本身的测试工作。



数据库功能性测试

■ 功能测试内容

- 安装与配置

- 测试数据库管理系统是否具有完整的图形化安装程序
- 是否提供集中式多服务器管理及网络配置
- 是否在安装界面中显示数据文件、日志文件、控制文件等参数文件的默认路径及其命名规则
- 是否提供运行参数查看与设置功能，能够正确地进行数据库的创建和删除

- 数据库存储管理

- 表空间（文件组）管理、数据文件管理、日志文件管理以及归档文件管理



数据库功能性测试

- 模式对象管理

数据库管理系统最基本的数据管理服务功能特性，是数据库所有功能的基础

- 表管理
- 索引管理
- 视图管理
- 约束管理
- 存储过程管理
- 触发器管理



数据库功能性测试

- 非模式对象管理
 - 模式管理
 - 模式的创建、删除、查看、用户指派
 - 用户管理
 - 用户的创建、删除、修改、授权、口令策略管理
 - 角色管理
 - 角色的创建、删除、修改、查看、用户指派
 - 权限管理
 - 数据库对象权限的查看与指派、用户对象权限的查看与指派
 - 审计选项设置
 - 语句审计、对象审计、权限审计、审计开关等
- 交互式查询工具
 - 易用性、稳定性



8.2.2 数据库功能性测试

- 性能检测与调优
 - 要求以图形方式提供SQL语句执行计划
 - 提供数据库运行图形监控
 - 提供可配置的性能数据跟踪与统计
 - 提供死锁监测与解锁功能
- 数据迁移工具
 - 支持txt文件的数据迁移
 - 支持excel文件的数据迁移
 - 支持XML数据导出
 - 支持从SQLServer的表、约束及数据迁移
 - 支持从Oracle的表、约束及数据迁移
 - 支持从DB2的表、约束及数据迁移
 - 支持从Oracle进行数据迁移的性能
- 作业管理

包括作业调度、通知（操作员）管理、维护计划管理等



数据库功能性测试

■ 测试方法

- 采用黑盒测试方法，可以通过图形化管理工具、交互式SQL工具等对数据库管理系统的功能特性进行测试。
- 要求被测数据库提供Windows和Linux平台上的图形化管理工具，任一平台上的工具都能够管理Windows和Linux平台上的数据库服务器。
- DataFactory 工具

数据库性能测试及原因分析

■ 数据库性能测试

- 一般来说，引起数据库性能问题主要原因有两个：数据库的设计和SQL语句。
- 数据库的性能优化可以从以下方面考虑：
- 物理存储
- 逻辑设计
- 数据库的参数调整
- SQL语句优化
- 可以借助一些工具来帮助找出有性能问题的语句

■ 数据库性能问题及原因分析

- 数据库服务器性能问题主要表现在某些类型操作的响应时间过长，同类型事务的并发处理能力差和锁冲突频繁发生等方面。
- 单类型事务响应时间过长
- 并发处理能力差
- 锁冲突严重
- 性能瓶颈的处理方法

性能瓶颈的处理方法

- 解决数据库性能问题是一个迭代和反复的过程，通常需要在各种条件的矛盾之间寻求最佳的平衡点。
 - 监视并记录性能相关数据
 - 重点监视数据库服务器软件性能属性
 - 定位资源占用较大的事务并做出必要的优化或调整
 - 将批任务放置在系统具有充分空闲时间时进行
 - 定位锁冲突，修改锁冲突发生严重的应用逻辑
 - 进行必要的数据库分布
 - 通过数据库服务器的并行执行特性，使得单一事务的执行具有较短的响应时间和不同类的事务之间影响相对缩小。
 -

数据库可靠性及安全性测试

■ 可靠性测试

数据库系统的稳定可靠性是应用企业最关心的问题，它与整个企业的经营活动密切相关，是支撑企业应用的后台核心和基础，测试点如下

- 数据库备份
- 故障恢复
- 运行稳定性
- 数据库复制

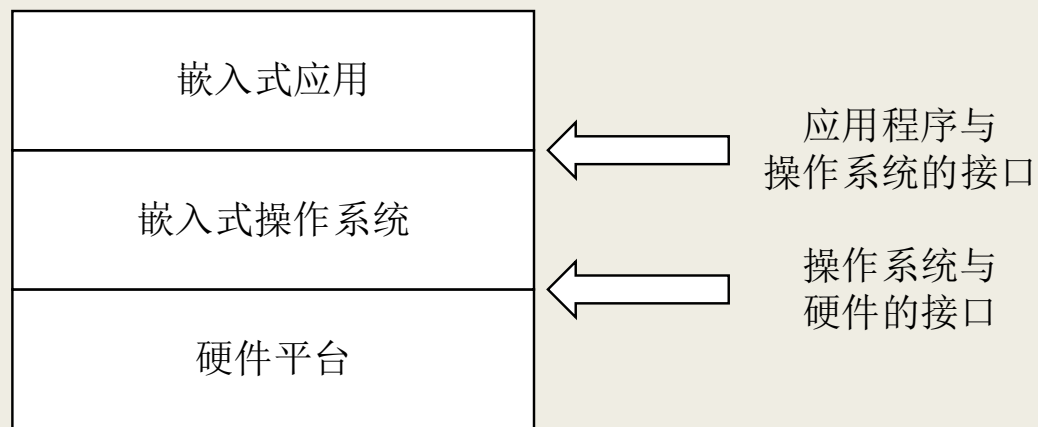
■ 安全性测试

数据库的安全性主要是指数据库的用户认证方式受其权限管理、当数据库遭受非法用户访问时，系统的跟踪与审计功能等，测试点如下：

- 用户及口令管理
 - 包括用户定义与管理、角色定义与管理、口令管理等
- 授权和审计管理
 - 主要测试点为数据库审计、授权管理（表权限/列权限）、支持操作系统用户验证方式等

嵌入式系统测试

- 嵌入式系统由嵌入式硬件与嵌入式软件组成。
- 硬件以芯片、模板、组件、控制器形式嵌入到设备内部，软件是实时多任务OS和各种专用软件，一般固化在ROM或闪存中。
- 嵌入式系统主要用于各种信号处理与控制。



嵌入式系统的基本结构

测试策略及测试流程

■ 嵌入式软件测试问题及测试方法

嵌入式软件开发一般采用典型的“宿主机/目标机”交叉方式，即利用宿主机上丰富的资源及良好的开发环境，通过串口或网络等将交叉编译生成的目标代码传输并装载到目标机，用调试器在监控程序或实时内核/OS的支持下实时分析、测试和调试，目标机在特定的环境（如分布式环境）下运行。

- 嵌入式软件的测试也称为交叉测试 (cross-testing)
- 对实时嵌入式系统进行分析需要建模、仿真和数学计算
- 对嵌入式应用，无论是测试还是调试，有效的方法仍是借助硬件仿真或软件模拟的手段来进行软件的测试和调试。

测试策略及测试流程

■ 嵌入式软件的测试流程

- 1.使用测试工具的插装功能（主机环境）执行静态测试分析并且为动态覆盖测试准备好一插装好的软件代码；
- 2.使用源码在主机环境执行功能测试，修正软件的错误和测试脚本中的错误；
- 3.使用插装后的软件代码执行覆盖率测试，添加测试用例或修正软件的错误，保证达到所要求的覆盖率目标；
- 4.在目标环境下重复2，确认软件在目标环境中执行测试的正确性；
- 5.若测试需要达到极端的完整性，最好在目标系统上重复3，确定软件的覆盖率没有改变。

嵌入式软件测试代表工具

■ 嵌入式白盒测试工具

- 白盒测试以源代码为测试对象，除对软件进行通常的结构分析和质量度量等静态分析外，主要进行动态测试。

- Logiscope TestChecher、Test RealTime、CodeTest、VectorCAST

■ 嵌入式黑盒测试工具

- 以硬件方式将被测系统的输入/输出端口用硬件对接相连，使用实时处理机和宿主机对被测系统进行激励和输入，实施驱动，然后获取输出结果进行分析，进行开环或闭环测试。

- ADS-2系统、GESTE嵌入式系统测试环境

嵌入式软件测试代表工具

■ 嵌入式灰盒测试工具

- 灰盒测试是指嵌入式软件既能做白盒测试，又能做黑盒测试的测试工具。目前主要是基于全数字仿真或半实物仿真技术的应用。

■ SPACEBEL、SHAM、CRESTS/ATAT、CRESTS/TESS

■ 嵌入式软件仿真工具

- 空间飞行器卫星等工作在太空中，它们的控制软件，即嵌入式软件的调试与测试必须在个等价太空环境下的仿真环境里进行，仿真环境的建立需要仿真工具的支持。

■ Eurosim、SPINEware



游戏测试

■ 游戏开发与测试过程

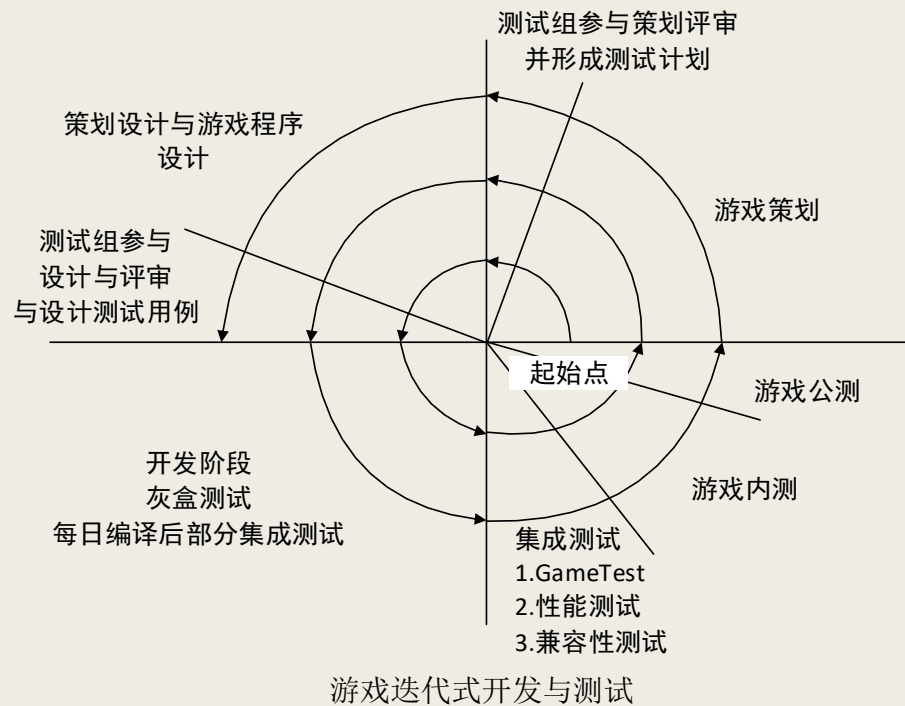
- 游戏开发必要条件

- Vision (设计)
- Technology (技术)
- Process (过程)

- 游戏测试与开发过程的关系

- 由于网络游戏的生命周期一般是三四年，因此常常采用迭代式的开发过程，既可以适应网络游戏本身这种长周期的开发，又可以充分利用RUP的迭代式开发的优点与CMM/CMMI框架中的里程碑控制来进行开发管理，从而达到对游戏产品的全生命周期的质量保证。

游戏开发与测试过程



测试的工作与游戏的开发是同步进行的，在每一个开发阶段中，测试人员都有参与，这样能够尽早地、深入地了解到系统的整体与大部分的技术细节，从而从很大程度上提高了测试人员对错误问题判断的准确性，并且可以有效地保证游戏系统的质量。



游戏开发与测试过程

- 与通用软件的开发过程区别

- 通用软件的需求明确，游戏软件的需求存在理想化
- 通用软件开发过程中需求变更少，游戏软件开发过程中需求变化快
- 开发过程的阶段不同



游戏测试主要内容

- 游戏测试可分为两部分：一是传统的软件测试，二是游戏本身的测试
- 游戏测试特性
 - 游戏情节的测试
 - 游戏世界中的任务系统组成，有人也称其为游戏世界的事件驱动，也可称为游戏情感世界的测试
 - 游戏世界的平衡测试
 - 主要表现在经济平衡，能力平衡（包含技能、属性等），保证游戏世界竞争公平
 - 游戏文化的测试
- 游戏可玩性测试

游戏测试的最重要内容，其本质是功能性测试

 - 游戏基本功能（任务）测试，保证游戏基本功能被覆盖。
 - 游戏系统虚拟世界的搭建，在构建交互平台的前提下进行游戏完整情节的系统级别的测试。
 - 游戏软件的风格、界面测试
 - 游戏性能、压力等必要的软件特性测试

可玩性测试是游戏最重要的一块，只有玩家的认同游戏才可能成功

游戏测试主要内容

- 但一些具体的涉及到某个数据的分析，比如PK参数的调整、技能的增加等一些增强可玩性的测试则需要职业玩家对它进行分析。
 - 内部的测试人员，他们都是精选的职业玩家分析人员，对游戏有很深的认识，在内部测试时，对上面的4点进行分析
 - 利用外部游戏媒体专业人员对游戏作分析与介绍，既可以达到宣传的效果，又可以达到测试的目的
 - 利用外部一定数量的玩家对外围系统的测试，主要测试游戏的可玩性与易用性，发现一些外围的缺陷
 - 游戏进入到最后阶段时，还要做内测、公测，让更多的人参与测试，测试大量玩家下的运行情况。

游戏测试的实施

■ 游戏策划与测试计划

在测试实施前，通过策划书来制定详细的测试计划

- 游戏程序本身的测试计划

- 比如任务系统、聊天、组队、地图等由程序来实现的功能测试计划

- 游戏可玩性测试计划

- 比如经济平衡标准是否达到要求，各个门派技能平衡测试参数与方法，游戏风格的测试

- 关于性能测试的计划

- 比如客户端的要求，网络版的对服务器的性能要求

■ 游戏性能测试

性能主要涉及以下几个方面

- 应用在客户端性能的测试

- 应用在网络上性能的测试

- 应用在服务器端性能的测试

网络版游戏，性能是至关重要的

游戏测试的实施

- 目前网络游戏主要分为传统的C/S架构的网络游戏、B/S架构的网络游戏和WAP网络游戏
 - C/S架构的网络游戏

需要用户下载客户端，然后通过客户端来访问服务器进行登录和游戏

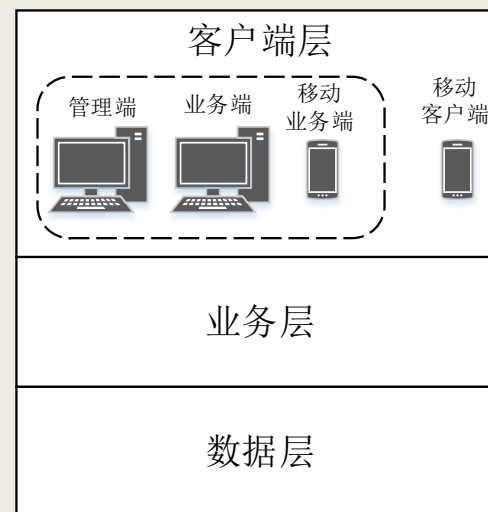
 - 自主研发一个机器人程序，模拟玩家登录与游戏
 - 使用性能测试工具进行性能测试
 - 进行封测、内测、公测等开放性测试方法
 - B/S架构的网络游戏

无须下载客户端，无须担心机器配置不够，就可以享受到网游的乐趣

 - 使用工具来模拟用户访问，与其他B/S架构的软件产品一样，通过各种工具和协议来模拟用户访问服务器，与服务器进行交互。
 - 进行封测、内测、公测等开放性测试方法
 - WAP网络游戏
 - 使用模拟器在电脑上模拟WAP环境，然后使用工具来进行性能测试。使用的协议可以使WAP，也可以是SOAP等其他协议。
 - 进行封测、内测、公测等开放性测试方法

移动应用软件测试

- 基于移动终端的便携性，移动应用呈现出跨平台、轻量化和Web化等特点。
- 移动应用类型涉及语音类、消息类、视频类、内容类、个人信息管理类、位置服务、电子商务、游戏类等。
- 一般的移动应用系统结构如右图所示
- 移动应用的测试类型涉及功能性测试、性能测试、安全性测试、稳定性测试、易用性测试、可靠性测试、兼容性测试及非技术性测试。
- 众多的移动应用及其推向市场的快速响应需求，以及移动终端使用的便利性，对测试的质量和响应速度提出了更高的要求。



移动应用测试的困难

- 移动开发和测试过程中，必须使用手机/终端真实环境进行系统测试
- 由于手机/终端操作系统的不同，以及操作系统版本之间的差异，使得真机系统测试这个过程尤其复杂
 - 手机/终端配置测试实验室
 - 安全问题严重
 - 专业测试队伍
 - 管理难度加大



移动应用测试类型

App测试就是要找出App中的Bug，通过各种手段和测试工具，判断App系统是否能够满足预期标准。

■ 冒烟测试(SmokeTesting)

- 测试的对象是每一个新编译的、需要正式测试的App版本，目的是确认软件基本功能正常，可进行后续的正式测试工作。
- 如果通过了该测试，则可以根据正式测试文档进行正式测试。否则，就需要重新编译，再次执行确认测试，直到成功。

■ 图形用户界面测试

- 用于核实用户与App之间的交互
- 主要测试在不同分辨率下，用户界面（如菜单、对话框、窗口和其他可视控件）布局、风格是否满足客户要求，文字是否正确，页面是否美观，文字、图片组合是否完美，操作是否友好等。

■ 安全性测试

- 移动设备/智能终端的安全性是一个需要考虑的重大问题
- 测试包括应用程序级别和系统级别两个方面

移动应用测试类型

■ 性能测试

- 测试移动App在真实环境中的运行性能，以及与硬件、网络资源的匹配度，最终度量系统相对于预定义目标的差距。
- 性能测试主要采取以下测试方法
 - 负载测试、强度测试、稳定性测试、基准测试、资源竞争测试和故障转移和恢复测试等

■ 兼容性测试

- 即配置测试，核实测试对象在不同的App、硬件配置中的运行情况，测试系统在各种软硬件配置、不同的参数配置下具有的功能和性能。

■ 网络测试

- 在网络环境下和其他设备对接，进行系统功能、性能与指标方面的测试，保证设备对接正常。

■ 本地化测试

- 为各个地方开发产品的测试，如英文版、中文版等，包括程序是否能够正常运行，界面是否符合当地习俗，快捷键是否正常起作用等，特别测试在A语言环境下运行B语言版本的App，看显示是否正常。

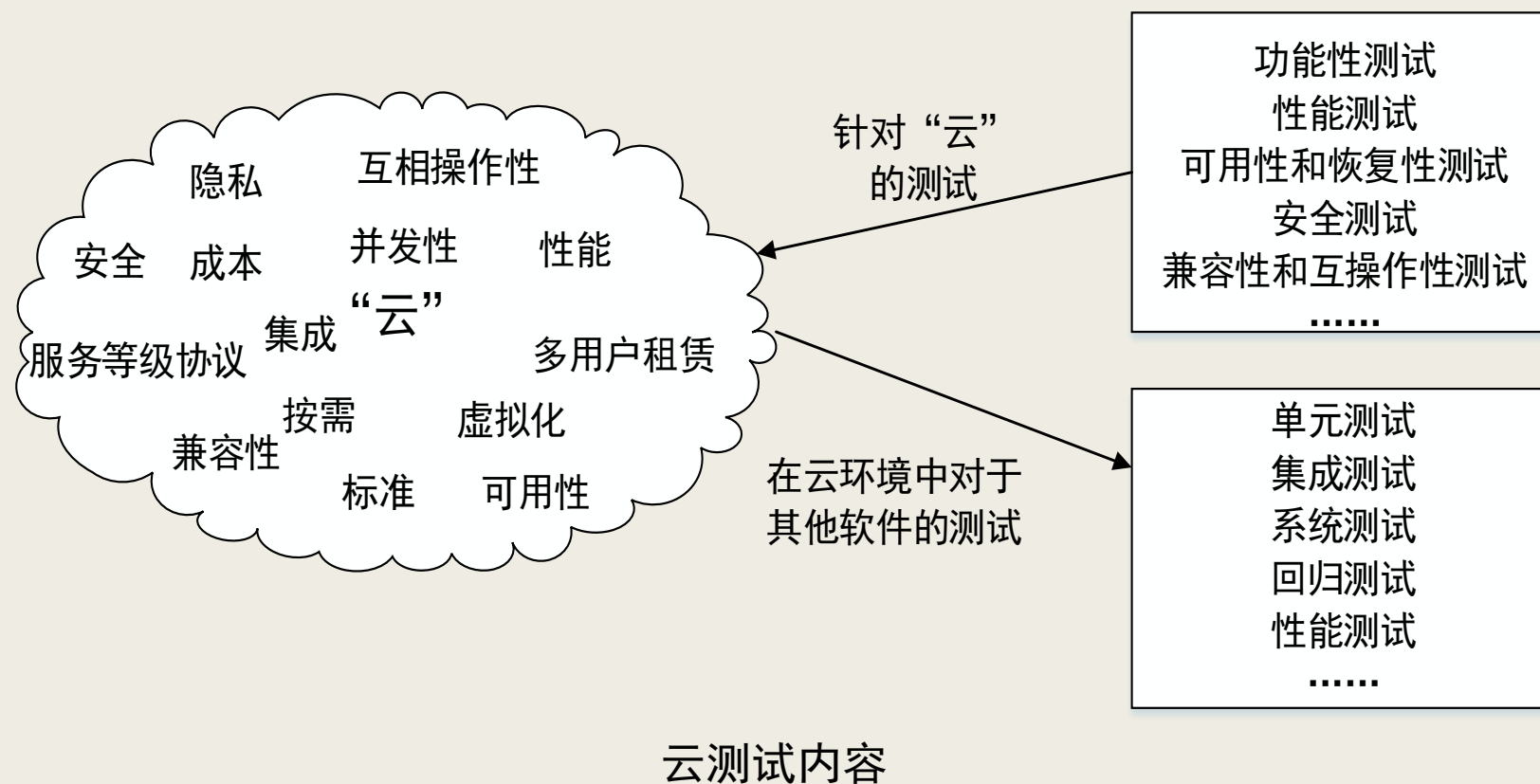


云应用软件测试

- 云计算的基本服务模式包括：软件即服务（Software as a Service, SaaS）、平台即服务（Platform as a Service, PaaS）、基础设施即服务（Infrastructure as a Service, IaaS），这些服务模式的出现改变了软件产品的生产和消费方式，软件测试的方法、技术和工具也需要随之变化
- 在云计算环境下，将软件测试过程迁移到云中，应用云计算平台提供的计算和存储等资源进行各种测试活动，这是一种新型的软件测试方式，是云计算技术的一种新应用。
- 云测试是通过“云”而实施的一种软件测试
 - 云测试是一种有效利用云计算环境资源对于其他软件进行的测试，或是一种针对部署在“云”中的软件进行的测试。
 - “测试资源”的服务化
 - 软件测试本身以统一接口、统一表示方式实现为一种服务，用户通过访问这些服务，实现软件测试，而不用关注“测试”所使用的技术、运行过程、实现方式等。
 - “测试资源”的虚拟化
 - 云计算的虚拟化实现方式，为云测试的虚拟化提供了较大的便利，测试资源的虚拟化，使测试资源可以随用户的需求提供，动态延展

云测试方法和技术

■ 云环境中的测试和针对“云”的测试



云测试方法和技术

■ 云环境中的测试和针对“云”的测试

- 在云环境中的测试

- 利用云资源对其他的软件系统进行测试，涉及与云测试密切相关的资源调度、优化、建模等方面问题，以便为其他软件搭建廉价、便捷、高效的测试环境，加速整个软件测试的进程。
- 在这一类型的测试中，其他的软件可以是传统意义上的本地软件，也可以是“云”中的应用软件服务。
- 云计算作为一种可以快速获得的有效资源，已经参与到软件测试的各阶段中，云计算能够快速配置所需测试环境，此种转变必然会给传统测试方式带来变革。

- 针对“云”的测试

- 涉及云计算内部结构、功能扩展和资源配置等多方面测试问题
- 测试部署在云环境中的各种云计算软件
- 各层的云服务对一般服务用户是透明的，它由大量动态、异构、复杂的系统构建，并且随着业务需求的变化，系统还在不断更新和演化，这必然导致很多隐藏的误差不容易被发现。

云测试方法和技术-针对“云”的测试

■ 功能测试

- 主要包括单元、集成、系统测试等内容，确保开发的云服务功能能够满足用户需求

■ 性能测试

- 包括压力和负载测试，测试云服务的性能能否满足用户按需服务的要求

■ 可用性和恢复性测试

- 主要针对发生灾难性事件后，“云”中的数据能够在较短暂的时间内快速恢复，使得云服务的可用性较高

■ 安全测试

- 确保云服务中存储、流动数据的保密性、完整性
- “云”的安全是云服务能否使用的关键

■ 兼容性和互操作性测试

- 为了确保开发的云服务能够运行在不同的配置环境下，如不同的操作系统、浏览器、服务器等

云测试现状及挑战

■ 云测试的现状

- 测试人员利用云测试服务商提供的测试环境，运行自己的测试用例。
- 云测试服务商为测试人员提供测试执行的服务。测试人员编写好测试用例后，提交给云测试平台，云测试平台执行测试并返回测试结果。
- 测试中需要使用软件工具或测试运行于不同测试环境都可进行云测试。

■ 云测试挑战

- 数据安全
 - 用户数据都是基于云环境的，会涉及用户敏感数据的隐私问题；同时随着应用信息的交互，这些数据会在不同系统之间流动。
- 集成问题
 - 云计算软件系统是由多个异构系统构成的，提供用户不同的云计算服务，满足了用户需求，但也增加了系统的复杂性，而这些异构系统彼此间很难获得对方的代码，加大了集成测试的难度。
- 多用户租赁
 - 云平台上的云应用是多用户租赁环境下的应用系统，多个用户共享一个实例化的应用实体及数据达到个性需求的目的，这就要求用户能够正确完成自身的操作功能，而彼此间的并发操作不会产生相互影响。

云测试现状及挑战

- 服务保障

- 由于故障（如响应时间延长、网络带宽等）导致服务不可用的情况，会降低人们使用云服务的热情。

- 并发问题

- 云服务可以迅捷地提供测试其他软件所需的资源和环境，但并不是所有的测试过程和场景都适合云测试框架，需要考虑系统间、测试用例间相互的依赖关系。

- 兼容和交互性

- 云计算中的软件运行在多个不同环境中，那么测试比以往都要复杂，测试的环境显得更加不可控制，需要考虑“云”中软件 and 不同环境的兼容以及与其他“云”的兼容问题。

- 虚拟化问题

- 虚拟化技术提高了资源的利用效率，然而并不是所有的测试方案都支持虚拟化技术：同台机器上产生的多个虚拟设备存在资源的竞争机制，这样测试的结果可能会与实际有的偏差。