

# 软件质量保证与测试

Software Quality Assurance and Testing

## 第 6 章 系统测试



金陵科技学院

# 什么是系统测试

系统测试是将经过集成测试的软件，作为计算机系统的一个部分，与系统中其他部分结合起来，在实际运行环境下对计算机系统进行的一系列严格有效地测试，以发现软件中潜在的问题。

系统测试的对象不仅仅包括开发出来的软件，还包括软件运行所依赖的硬件和接口、操作系统、其它支持软件以及相关数据等。

# 什么是系统测试

系统测试的依据是软件的规格说明书，通过测试验证软件系统是否符合软件规格，找出与软件规格不符或与之矛盾的地方。

## 系统测试的主要测试技术

- ❖ 系统测试完全采用黑盒测试技术，因为这时已不需要考虑组件模块的实现细节，而主要是根据需求分析时确定的标准检验软件是否满足功能、行为、性能和系统协调性等方面的要求。

- 系统测试所用的数据必须尽可能地像真实数据一样精确和有代表性，也必须和真实数据的大小和复杂性相当。满足上述测试数据需求的一个方法是使用真实数据。
- 在不使用真实数据的情况下应该考虑使用真实数据的一个拷贝。拷贝数据的质量、精度和数据量必须尽可能地代表真实的数据。当使用真实数据或使用真实数据的拷贝时，仍然有必要引入一些手工数据。在创建手工数据时，测试人员必须采用正规的设计技术，使得提供的数据真正有代表性，确保软件系统能充分地测试。

## 系统测试人员

- ❖ 系统测试由独立的测试小组在测试组长的监督下进行，测试组长负责保证在合理的质量控制和监督下使用合适的测试技术执行充分的系统测试。在系统测试过程中，测试过程由一个独立测试观察员来监控测试工作。系统测试过程也应考虑邀请一个用户代表非正式地观看测试，同时得到用户反馈意见并在正式验收测试之前尽量满足用户的要求。

# 系统测试的分类

从用户  
的角度

I

用户支持测试

II

用户界面测试

III

可维护性测试

IV

安全性测试

# 系统测试的分类

从系统的角度

系统可靠性、  
稳定性测试

系统兼容  
性测试

系统安装  
升级测试

性能测试

系统组  
网测试



# 系统测试的分类

从业务功能的角度

基本功能测试：测试系统核心功能、基本业务流程。

业务场景测试：遍历测试主要业务流程和业务规则。

业务功能覆盖：测试软件规格定义的所有功能是否都已实现。

# 系统测试的分类

从业务功能的角度

业务功能组合：相关联的功能项的组合功能都被正确实现。

业务功能冲突：业务功能间存在的功能冲突情况均测试通过，例如：共享资源访问等。

异常处理及容错性：输入异常数据，或执行异常操作后，系统容错性及错误处理机制的健壮性。

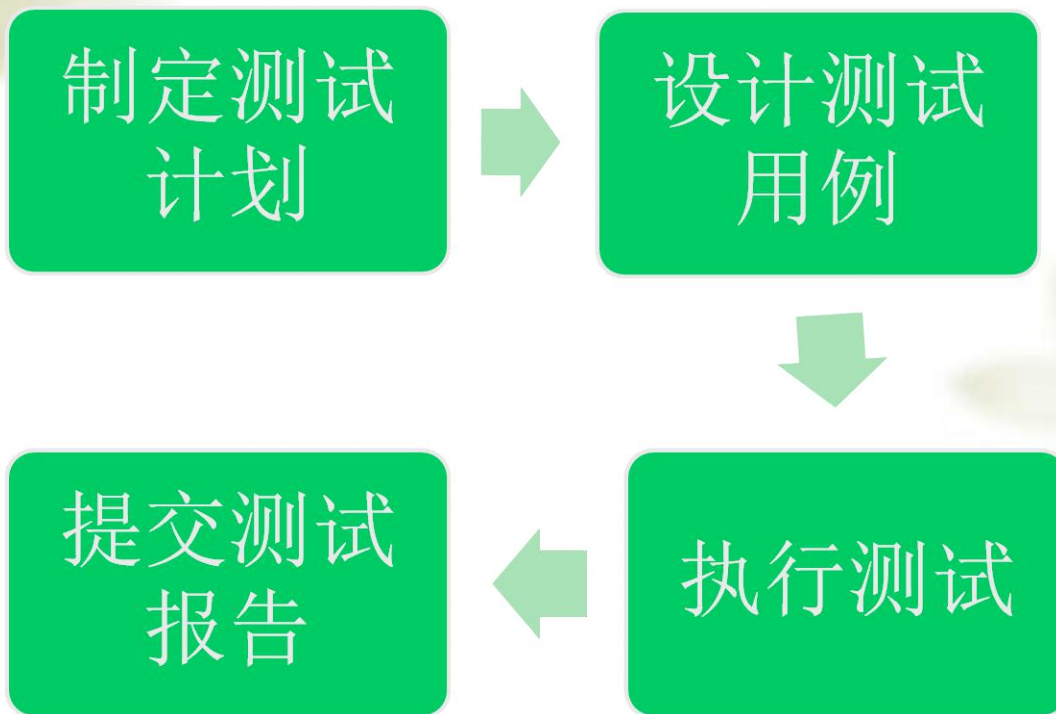
## 系统测试一般要完成以下几种测试

1. 功能测试
2. 性能测试
3. 系统可靠性、稳定性测试
4. 系统兼容性测试
5. 恢复测试
6. 安全测试
7. 强度测试
8. 面向用户支持方面的测试
9. 其他限制条件的测试

# 系统测试的分类

- ❖ 这么多种的系统测试项目，对某一个软件而言，并不是都必须要进行的，实际测试中根据软件的特点和实际需要，有所侧重的选用。
- ❖ 另外有些系统测试项目专业性很强，可以交由第三方测试机构进行，以发挥其专业技术优势和独立性优势，以进一步促进软件质量的提升。

# 系统测试的过程



## 1.安装与卸载测试

- ❖ 安装测试（Installing Testing）是确保软件在正常情况和异常情况下都能进行安装，并核实软件在安装后可立即正常运行的测试。
- ❖ 异常情况包括磁盘空间不足、缺少目录创建权限等场景。
- ❖ 在安装过程中，注意测试软件给用户的提示是否清楚明了、安装的操作是否容易、安装过程是否太冗长、各系统设置是否正确、安装完成后软件是否能正常运作、安装过程有没有干扰计算机中其他的程序等。



## 1.安装与卸载测试

- ❖ 卸载测试：卸载测试要考虑卸载过程中，系统的提示是否清楚明了、操作是否简单、卸载是否彻底、系统设置是否回复到安装前状态等。
- ❖ 软件卸载通常遇到的问题是卸载不彻底，比如安装时设立的文件夹没有清除、 里面的设置是否清理干净等。

## 2.兼容性测试

❖ 兼容性测试是针对测试中软件与其他软件之间，以及被测试的软件与不同硬件之间的兼容性进行的测试。兼容测试应包括以下这些：

- ① 操作系统兼容
- ② 硬件兼容
- ③ 软件兼容
- ④ 数据库兼容
- ⑤ 数据兼容



### 3.安全测试

- ❖ 安全测试（**Security Testing**）用于检验系统对非法侵入的防范能力。系统的安全必须能够经受各方面的攻击。
- ❖ 安全测试是在软件产品开发基本完成到发布阶段，对产品进行检验以验证产品符合安全需求定义和产品质量标准的过程。

### 3.安全测试

- ❖ 在安全测试的过程中，测试人员扮演非法入侵者的角色，采用各种方法试图突破系统的安全防线。
- ❖ 从理论上讲，只要给予足够的时间和资源，任何系统都可以侵入。因此，系统安全设计原则是将系统设计为想攻破系统而付出的代价应大于侵入系统之后得到的信息价值，使得非法侵入者无利可图。

### 3.安全测试

❖ 常见的非法入侵手段有以下3种。

- (1) 尝试通过外部手段截获或破译系统口令。
- (2) 使用甚至专门开发，能够攻击目标对象的软件工具来实施攻击，试图破坏系统的保护机制。
- (3) 故意引发系统错误，导致系统失败，并企图趁系统恢复时侵入系统。

### 3.安全测试

#### 安全测试的目的

- ① 提升IT产品的安全质量；
- ② 尽量在发布前找到安全问题予以修补降低成本；
- ③ 度量安全。
- ④ 验证安装在系统内的保护机制能否在实际应用中对系统进行保护，使之不被非法入侵，不受各种因素的干扰。

# 安全测试与通常测试区别

- ❖ 1.目标不同：测试以发现**BUG**为目标，安全测试以发现安全隐患为目标。
- ❖ 2.假设条件不同：测试假设导致问题的数据是用户不小心造成的，接口一般只考虑用户界面。安全测试假设导致问题的数据是攻击者处心积虑构造的，需要考虑所有可能的攻击途径。

# 安全测试与通常测试区别

- ❖ 3.思考域不同：测试以系统所具有的功能为思考域。安全测试的思考域不但包括系统的功能，还有系统的机制、外部环境、应用与数据自身安全风险与安全属性等。
- ❖ 4.问题发现模式不同：测试以违反功能定义为判断依据。安全测试以违反权限与能力的约束为判断依据。

## 4.恢复测试

- ❖ 恢复测试是指：人为使系统发生灾难（系统崩溃、硬件损坏、病毒入侵等），检查系统是否能恢复被破坏的环境和数据。
- ❖ 恢复测试需要采取各种人工干预方式，强制性地使软件系统不能正常工作，再来检验系统能不能恢复到正确的状态。

## 4.恢复测试

- ❖ 恢复测试主要检查系统的容错能力。当系统出错时，能否在指定时间间隔内重新启动并恢复到正确的状态。
- ❖ 在这种测试中，将把应用程序或系统置于极端条件下（或者是模拟的极端条件下），以产生故障，然后调用恢复进程，并监测和检验应用系统，核实应用系统和数据是否已得到了正确的恢复。



## 4.恢复测试

- ❖ 恢复测试主要检查系统的容错能力。当系统出错时，能否在指定时间间隔内重新启动并恢复到正确的状态。
- ❖ 恢复测试首先要采用各种办法强迫系统失败，然后验证系统是否能尽快恢复。
- ❖ 对于自动恢复需验证重新初始化（reinitialization）、检查点(checkpointing mechanisms)、数据恢复(data recovery)和重新启动 (restart)等机制的正确性；对于人工干预的恢复系统，还需估测平均修复时间，确定其是否在可接受的范围内。

## 4.恢复测试

恢复测试中需要考虑的典型问题如下：

- （1）某种条件导致的故障，其后果是怎样的。
- （2）故障出现后，系统能否恢复到故障前正确的状态。
- （3）恢复的机制和过程是否可靠。
- （4）恢复过程所需要的时间和成本是否在可承受的范围之内。

## 5.性能测试

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。

并发测试是一种性能测试，主要测试当有多个用户并发访问同一个应用、模块或者数据时是否会产生隐藏的并发问题，如内存泄漏、线程锁、资源争用问题，几乎所有的性能测试都会涉及并发测试。

## 5.性能测试

负载测试和压力测试也都属于性能测试，两者可以结合进行。

通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

## 5.性能测试

### 性能测试应进行的工作

- (1) 对软件系统架构的分析（了解输入、输出数据类型、数据量）。
- (2) 网络环境说明。
- (3) 硬件环境说明。
- (4) 测试范围、目的与方法。
- (5) 性能测试工具的选型。
- (6) 测试的启动和退出条件。
- (7) 测试执行及测试结果分析。

## （1）影响系统性能的原因

- ❖ 网络状况
- ❖ 硬件设备
- ❖ 系统/应用服务器/数据库配置
- ❖ 数据库设计和数据库访问实现
- ❖ 业务的程序实现（算法）

## （2）性能测试目的

### ❖ 评估系统的能力：

☞ 测试中得到的负荷和响应时间数据可以被用于验证所计划的模型的能力，并帮助作出决策。

### ❖ 识别体系中的弱点：

☞ 受控的负荷可以被增加到一个极端的水平，并突破它，从而修复体系的瓶颈或薄弱的地方。

### ❖ 系统调优：

☞ 重复运行测试，验证调整系统的活动得到了预期的结果，从而改进性能。

## （2）性能测试目的

### ❖ 检测软件中的问题：

✎长时间的测试执行可导致程序发生由于内存泄露引起的失败，揭示程序中的隐含的问题或冲突。

### ❖ 验证稳定性（**resilience**）可靠性（**reliability**）：

✎在一个生产负荷下执行测试一定的时间是评估系统稳定性和可靠性是否满足要求的唯一方法。



### (3) 性能测试的时机

- ❖ 性能测试是在系统相对稳定时进行的，  
所以首先要进行功能测试
- ❖ 完成系统的集成测试，系统试运行阶段
- ❖ 系统运行时出现性能问题

## (4) 不同视角的系统性能

- ❖ 用户视角

- ☞ 响应时间

- ☞ 稳定性

- ❖ 系统视角

- ☞ 延迟（网络延迟、数据库延迟）

- ☞ 系统资源使用状况

- ❖ 开发者视角

- ☞ 代码实现

- ☞ 数据库实现

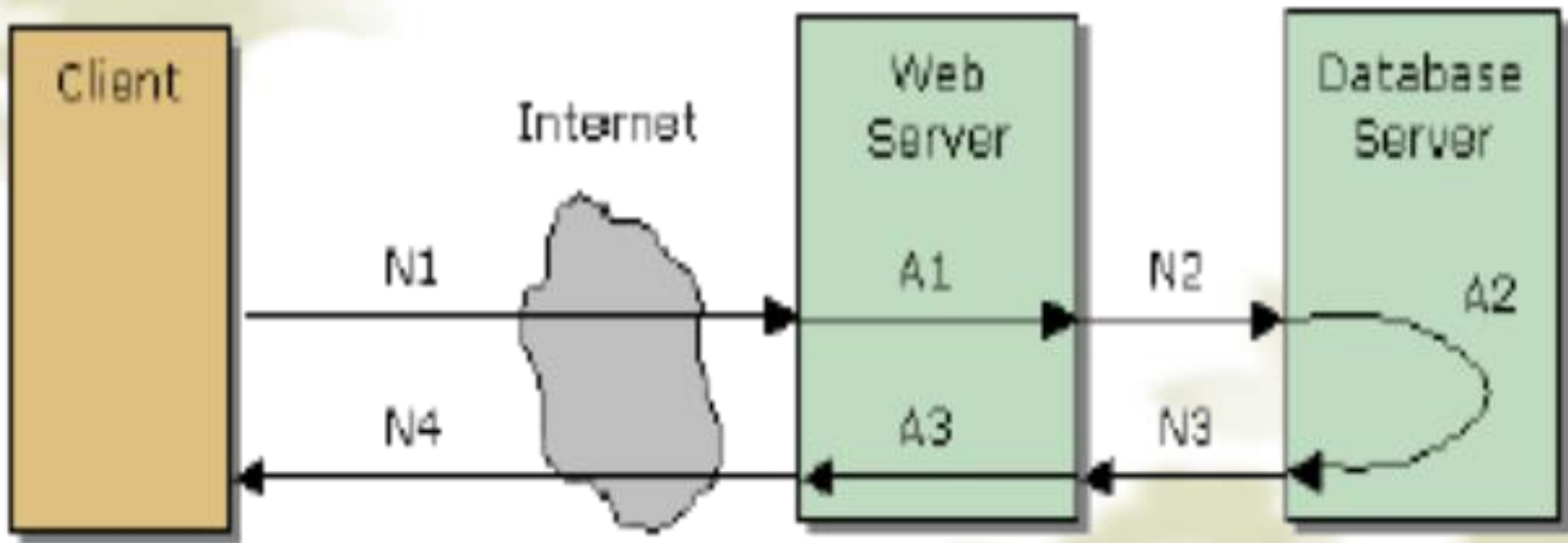
## （5）性能测试中的基本概念

- ❖ 响应时间（Response Time）
- ❖ 点击数（Hits）
- ❖ 页面请求（Page view）
- ❖ 吞吐量（Throughout）  
并发用户数\*（HTML文档大小）/请求时间
- ❖ 并发用户（Concurrency User）
- ❖ 资源利用率（Resource Usage）

## ❖ 通用指标

- ❑ **ProcessorTime**: 指服务器CPU占用率，一般平均达到70%时，服务就接近饱和；
- ❑ **Memory Available Mbyte** : 可用内存数，如果测试时发现内存有变化情况也要注意，如果是内存泄露则比较严重；
- ❑ **Physicsdisk Time** : 物理磁盘读写时间情况；

- 响应时间的分解（WEB应用）



响应时间=网络响应时间+ 应用程序响应时间

响应时间=(N1+N2+N3+N4)+(A1+A2+A3)

# 统计调查结果

- ❖ 事务响应时间不超过4秒，可以接受
- ❖ 事务响应时间大于4秒小于8秒，30%用户会撤消事务
- ❖ 事务响应时间大于8秒小于15秒，60%用户会撤消事务
- ❖ 事务响应时间超过15秒，超过90%用户会撤消事务

## (7) 响应时间类型

- 一般响应时间
  - 通常为系统更新、查询、报告等行为定义
  - 选择一些事务的实例来测试
- 特殊响应时间
  - 分别定义
  - 需要逐项测试

## (7) 响应时间类型

- 术语

- 连接时间：客户机和服务器建立连接的时间
- 发送时间：客户机向服务器发送数据的时间
- 接受时间：服务器向客户机发送响应数据的时间
- 处理时间：服务器向应客户机请求所需要的时间
- 事务时间：客户机处理接收数据需要的时间



## （8）性能测试的分类

性能测试类型包括：

- ❑ 负载测试：确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。
- ❑ 强度测试：强度测试是一种性能测试，在系统资源特别低的情况下软件系统运行情况。

## （8）性能测试的分类

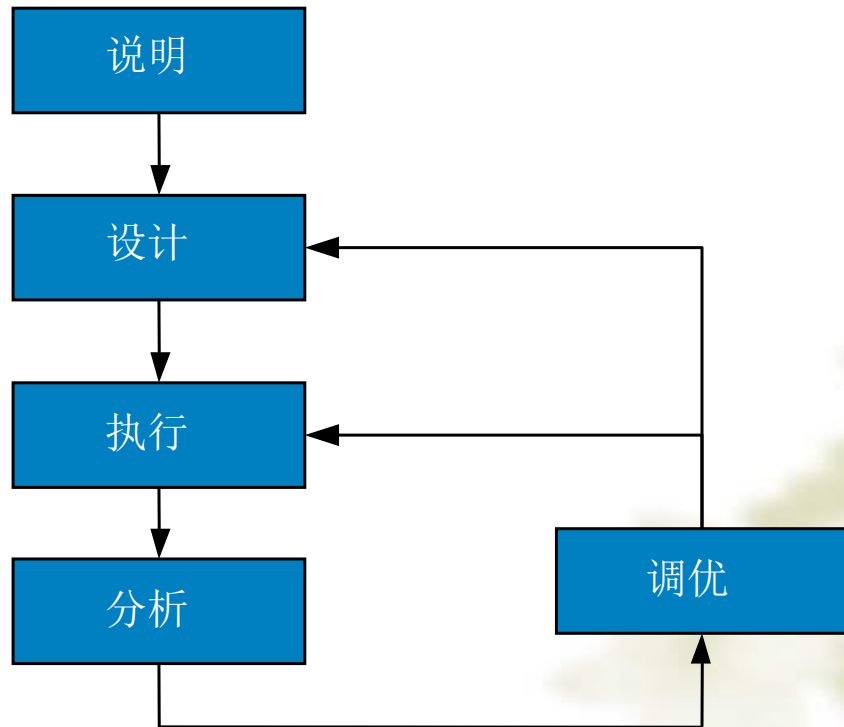
性能测试类型包括：

- ▣ 容量测试：确定系统可处理同时在线的最大用户数（在用户可接收的范围内）。
- ▣ 压力测试：考察当前软硬件环境下系统所能承受的最大负荷并帮助找出系统瓶颈所在。

## （9）性能测试的测试用例设计

- ❖ 预期性能指标的测试用例
- ❖ 并发用户相关的测试
- ❖ 疲劳强度与大数据量测试
- ❖ 网络性能测试
- ❖ 服务器性能测试

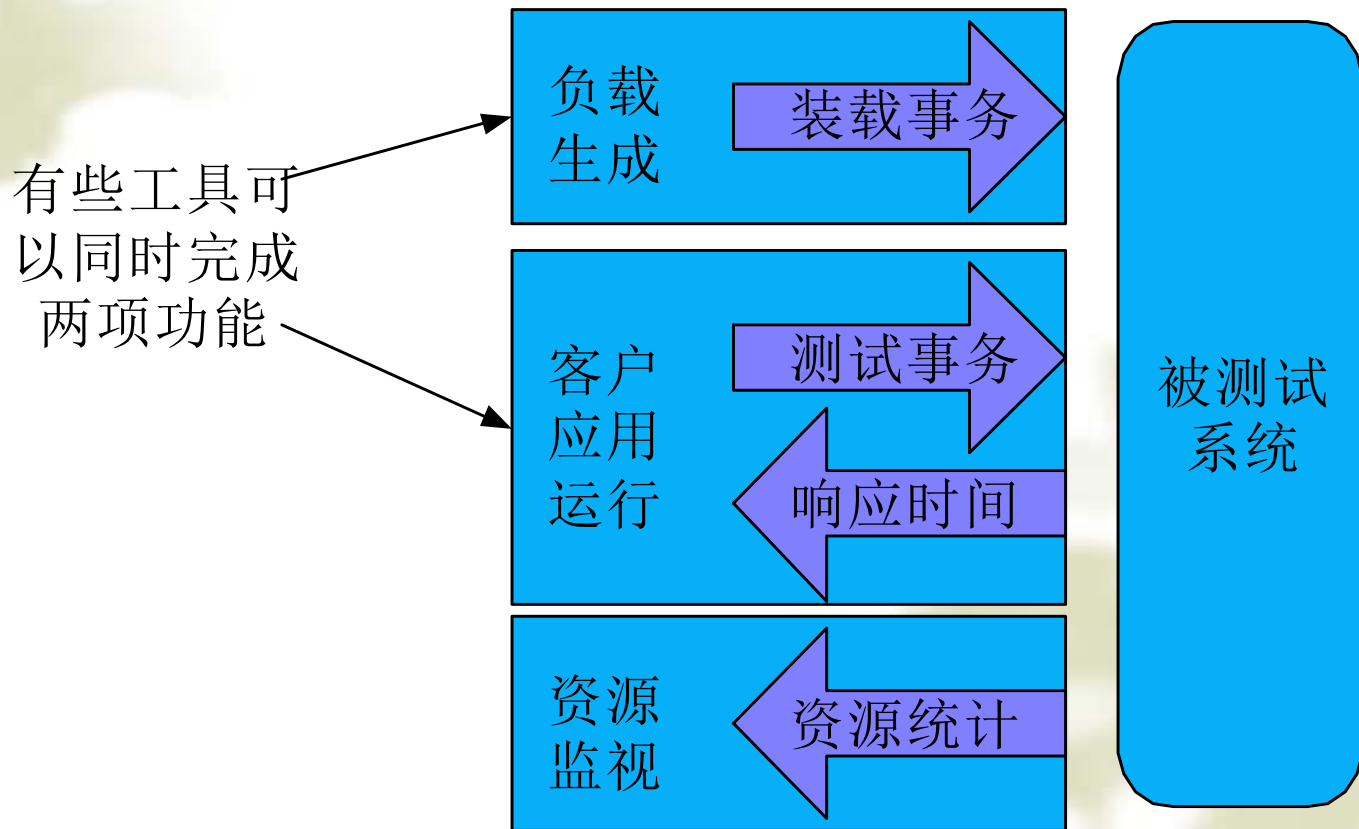
## (10) 性能测试的过程



## (11) 系统调优

- 包括对应用软件、中间件、数据库等的多项改变  
数据库调优的效果大于程序调优
- 服务器系统参数的改变
- 升级客户端、服务器硬件、网络性能或路由等

## (12) 性能测试的工具



# 测试工具一览表

	Rational	MI	Compuware	Segue	Empirix
测试管理	TestManager	TestDirector	Reconcile, QADirector, TrackRecord	SilkPlan Pro, SilkRadar	
Web测试	Rational Suite TestStudio, Rational Robot	Astra QuickTest, WinRunner, Quick Test Professional	QARun, DevPartner Java Edition, Web Check	SilkTest, SilkTest Interactional	e-test suite
功能测试	Rational Suite TestStudio, Rational Robot	Astra QuickTest, WinRunner, Quick Test Professional, XRunner	QARun	SilkTest, SilkTest Interactional, SilkPilot	e-tester
性能测试	Rational Team Test	Astra LoadTest, LoadRunner,Active Test,LoadRunner TestCenter	QALoad, QACenter Performance Edition	SilkPerformer	e-load



## 6.1 并发测试

并发测试是一种性能测试，主要测试当有多个用户并发访问同一个应用、模块或者数据时是否会产生隐藏的并发问题，如内存泄漏、线程锁、资源争用问题，几乎所有的性能测试都会涉及并发测试。

## 6.1 并发测试

- ❖ 一般只需针对软件容易出现并发、使用频繁的核心功能模块进行并发测试。
- ❖ 测试系统在高并发的情况下，会不会出现问题，能不能稳定运行，以及能否保持较好的响应速度。

## 6.1 并发测试

- ❖ 测试目的一方面是为了获得确切的性能指标，另外一个重要的方面就是为了发现并发引起的问题。
- ❖ 在具体的性能测试工作中，并发用户往往都是借助工具来模拟的，例如LoadRunner性能测试工具中叫做虚拟用户；因为实际情况中去实现同时多人并发的测试环境要求比较高而测试成本高、测试时间也是比较长。

## 6.1 并发测试

- ❖ 测试目的一方面是为了获得确切的性能指标，另外一个重要的方面就是为了发现并发引起的问题。
- ❖ 在具体的性能测试工作中，并发用户往往都是借助工具来模拟的，例如LoadRunner性能测试工具中叫做虚拟用户；因为实际情况中去实现同时多人并发的测试环境要求比较高而测试成本高、测试时间也是比较长。

## 6.1 并发测试

- ❖ 确定用户并发数：需要知道系统所承载的在线用户数；例如用户的总量、用户平均在线数值、用户最高峰在线数值。
- ❖ 例如：  
公司OA系统账号或者总用户有2000人；  
最高峰在线500人。

## 6.1 并发测试

但是最高峰在线500人，并不等于最多会有500个并发用户，即在线人数 不等于 并发人数。

500人中，可能有40%只是在浏览公司首页新闻、公告板之类，40%用户打开了公司OA系统，但没有进一步的操作，这两类操作几乎不对服务器产生持续的压力；另外20%用户在进行业务流程操作，如查询、操作表格等。

在这种情况下，只有后面的20%用户在对服务器造成实质性的影响。

## 6.1 并发测试

如果我们把查询、操作表格作为一个业务，那么可以直接将这部分业务并发用户称为并发用户数。

关于并发用户数有两个公式：

- ① 计算平均并发用户数： $C = NL/T$
- ② 并发用户峰值数： $C' \approx C + 3\sqrt{C}$

## 6.1 并发测试

公式（1）中， $C$ 是平均的并发用户数； $n$  是 login session的数量； $L$ 是login session的平均长度； $T$ 指考察的时间段长度。

公式（2）则给出了并发用户数峰值的计算方式中，其中， $C'$  指并发用户数的峰值， $C$ 就是公式（1）中得到的平均的并发用户数。该公式的得出是假设用户login session的产生符合泊松分布而估算得到的。



## 6.1 并发测试

### 实例：

假设有一个OA系统，该系统有3000个用户，（可以看注册信息）平均每天大约有400个用户要访问该系统（可以通过日志文件查看）。

对一个典型用户来说，一天之内用户从登录到退出该系统的平均时间为4小时，在一天的时间内，用户只在8小时内使用该系统。

## 6.1 并发测试

实例：

则根据公式（1）和公式（2），可以得到：

平均并发用户数： $C = 400 * 4 / 8 = 200$

并发用户峰值数： $C' \approx 200 + 3 * \sqrt{200} = 242$

## 6.1 并发测试

也有一个简单的做法是把每天访问系统用户数的10%作为平均的并发用户数。

最大的并发用户数可在平均并发用户数上乘一个值2或者3。

## 6.1 并发测试

假如某软件系统的规格说明要求每秒最大可以处理100个登录请求，那么可以模拟100个并发用户来执行登录测试，然后观察系统响应时间是否还在允许范围内，即可知系统是否达到并发规格要求。

## 6.2 负载测试

负载测试也属于性能测试，通过负载测试，可以确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。

负载测试是模拟实际软件系统所承受的负载条件的系统负荷，通过不断加载（如逐渐增加模拟用户的数量）或其它加载方式来观察不同负载下系统的响应时间和数据吞吐量、系统占用的资源（如CPU、内存）等，以检验系统的行为和特性，以发现系统可能存在的性能瓶颈、内存泄漏、不能实时同步等问题。

## 6.2 负载测试

### 一次加载

一次加载每个数量的用户，在预定的时间段内持续运行。例如，早晨上班，用户访问网站或登录网站的时间非常集中，基本上属于**Flat**负载模式。

### 递增加载

有规律地逐渐增加用户，每几秒增加一些新用户，交错上升，这种方式又称为**ra-mp-up**模式。借助这种负载方式的测试，容易发现性能的拐点，即性能瓶颈的位置。

## 6.2 负载测试

### 高低突变加载

某个时间用户数量很大，突然降级到很低，然后，过一段时间，又突然加到很高，反复几次。借助这种负载方式，容易发现资源释放、内存泄漏等问题。

### 随机加载方式

由随机算法自动生成某个数量范围动态的负载。虽然不容易模拟系统运行出现的高峰期，但能模拟长时间的高位运行过程。

## 6.3 压力测试

- ❑ 压力测试可以被看作是负载测试的一种，即高负载下的负载测试。
- ❑ 压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。
- ❑ 压力测试是持续不断地给被测系统增加压力，直到被测系统被压垮，从而确定系统能承受的最大压力。
- ❑ 通过压力测试我们可以知道系统能力的极限，有时需要找出系统瓶颈之所在，然后改进系统，提升系统能力



## 7.疲劳强度与大数据量测试

- ❖ 疲劳测试目的：确定系统处理最大工作量时的性能。
- ❖ 内容：在系统稳定运行下，模拟最大或者恰当的负载、长时间运行系统（经验上一般是连续**72**个小时），通过综合分析执行指标和资源监控来分析系统的稳定性，确定系统处理最大工作量强度时的性能指标和变化过程。

## 7.疲劳强度与大数据量测试

❖ 大数据量测试分为两种：

➤ 独立的数据量测试

针对某些系统存储、传输、统计、查询等业务进行大数据量测试

➤ 综合数据量测试

和压力性能测试、负载性能测试、疲劳性能测试相结合的综合测试方案

## 7.疲劳强度与大数据量测试

- ❧ 针对某些系统存储、传输、统计查询等业务进行大数据量的测试,可以通过工具完成;
- ❧ 数据库性能测试,可以编写程序和测试工具结合实现;
- ❧ 是与前面并发测试相结合的组合数据访问的大数据量测试。编写用例时主要编写前一部分,后一部分尽量放在并发相关的测试用例中。

## 8.网络性能测试

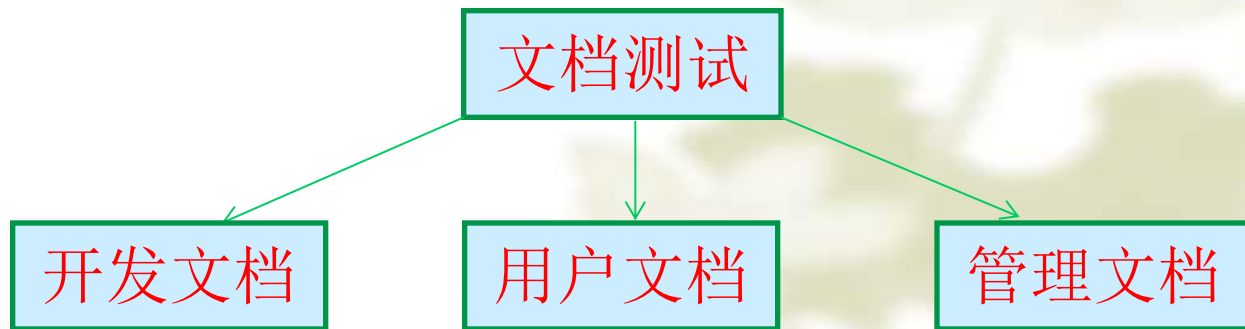
- ❖ 主要是为了准确展示带宽、延迟、负载和端口的变化是如何影响用户的响应时间的。
  - ✎ 测试用户数目与网络带宽的关系。
  - ✎ 用例的编写应和用户并发测试结合起来:例如在原有的基础上采用工具来调整网络设置。

## 9.服务器性能测试

- 测试用例不必独立编写，也可以根据实际需要编写少量的测试用例。系统集成部会完成一部分工作。
- 建议：这部分和前两部分的用例编写结合起来，在用户并发性能测试、疲劳强度与大数据量性能测试时完成对服务器性能的监控。
- 软件开发设计人员在软件开发设计时，不可能完全预见用户实际使用软件系统的情况。
- 另外，一个软件产品可能拥有众多用户，不可能由每个用户验收，此时多采用称为  $\alpha$  、  $\beta$  测试的过程，以发现那些似乎只有最终用户才能发现的问题。

## 10.文档测试

- ❖ 软件文档分为三类：用户文档、开发文档、管理文档
- ❖ 文档测试：检验文档的完整性、正确性、规范性、一致性、易理解性、易浏览性。
- ❖ 文档是软件的一部分，很重要。



## 11.界面测试

- ❖ 界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。
- ❖ 而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。

# GUI测试内容

## ❖ 窗口（1）

- ❧ 窗口能否基于相关的输入或菜单命令适当地打开；
- ❧ 窗口能否改变大小、移动和滚动；
- ❧ 窗口中的数据能否用鼠标、功能键、方向箭头和键盘操作；
- ❧ 当被覆盖并重新调用后，窗口能否正确再生；
- ❧ 需要时能否使用所有窗口相关的功能，所有相关的功能是否可操作；
- ❧ 是否有相关的下拉式菜单、工具条、滚动条、对话框、按钮、图标和其它控制，既正确显示又完全可用；



# GUI测试内容

## ❖ 窗口（2）

- ❧ 显示多个窗口时，窗口的名称能否正确表示，活动窗口是否被加亮；
- ❧ 如果使用多用户，所有窗口是否能被实时更新；
- ❧ 多次或不正确按鼠标是否会产生无法预料的结果；
- ❧ 窗口的声音和颜色提示和窗口的操作顺序是否符合需求；
- ❧ 窗口能否正确被关闭。

# GUI测试内容

- ❖ 下拉式菜单和鼠标操作（1）
  - ❧ 菜单条是否显示在合适的语境中；
  - ❧ 应用程序的菜单条是否显示系统相关的特性；
  - ❧ 下拉式操作能否正确进行，功能是否正确；
  - ❧ 菜单、调色板和工具条能否正确的工作；
  - ❧ 能否正确列出所有的菜单功能和下拉式子功能；

# GUI测试内容

## ❖ 下拉式菜单和鼠标操作（2）

- ❧ 能否通过鼠标操作所有的菜单功能，能否通过用其它的文本命令激活每个菜单功能；
- ❧ 菜单功能能否随当前的窗口操作加亮或变灰；
- ❧ 如果要求多次点击鼠标，或鼠标有多个按钮，能否正确识别；
- ❧ 光标、处理指示器和识别指针能否随操作而适当改变。

# GUI测试内容

## ❖ 数据项

- ❧ 字母数字数据项能否正确回显，并输入到系统中；
- ❧ 图形方式的数据项（如滚动条）是否正常工作；
- ❧ 数据输入消息是否可理解，能否识别非法数据。

## ✓ 提示信息

- 报信对于通知用户出错的类型，为用户提供控制输入顺序和修改错误是很重要的。
- 在报信时所用的行文应当用词准确、简明、完备。
- 出错信息的提示应当报告错误出在何处，是什么错误，为什么错了，以及要修改错误应当采取什么措施。
- 提示信息不应使用专业术语，应当使用肯定方式和主动语态。

## 其它数据输入的方法

- 光学标记 / 识别 (OMR)

光学标记 / 识别在表格中使用。用户在表格的一个区域中打标记 ☐ 或 ☒，然后让表格通过一个光敏读入设备，其中用暗标记 ☒ 表示“是”，用亮标记 ☐（即未标记过）表示“否”。

- 光学字符识别 (OCR)

OCR系统可让计算机通过模式比较来识别一些具有不同字体和大小的印刷体。首先它让字符识别系统熟悉铅字字体的特征。

- 磁性墨水字符识别 (MICR)  
MICR字体就是在银行支票上的帐号和分类号所使用的字符。
- 条形码 (Bar Code)  
条形码由许多粗细不等的竖线组成的标签，这些竖线条在特定位置上出现或不出现就表示某个特定的数据。
- 声音数据输入  
声音数据输入有许多很明显的优点。它输入速度很快，可用于不宜使用纸张及不能使用键盘的场合。这种方式不需要书写，只需用户使用自己的声音器官发声即可。

# 集成测试与系统测试的区别

## 1、测试对象不同：

集成测试的测试对象只是软件本身。而系统测试的测试对象，是包括软件及其运行环境组成的整个系统。

## 2、测试依据不同

集成测试的依据是系统概要设计书。系统测试的依据是软件规格说明书。



# 集成测试与系统测试的区别

## 3、测试时间不同

在测试时间上，集成测试先于系统测试。集成测试介于单元测试和系统测试之间。

## 4、测试方法不同

集成测试通常会采用白盒+黑盒测试。而系统测试通常使用黑盒测试。

## 集成测试与系统测试的区别

### 5、测试内容侧重点不同

集成测试的侧重点是各个单元模块之间的接口，以及各个模块集成后所实现的功能。而系统测试的主要内容就是整个系统的功能和性能。

### 6、测试人员不同

集成测试工作一般由开发人员或者开发人员和测试人员的共同完成。系统测试一般由测试人员完成。