



SOFTWARE TESTING

苏临之

sulinzhi029@nwu.edu.cn



Common Issues in System Testing

- Ability Testing
- Load Testing
- Stress Testing
- Performance Testing
- Memory Testing



Specific Issues in System Testing

- Configuration Testing
- Compatibility Testing
- Foreign Language Testing
- Usability Testing
- Documentation Testing
- Safety Testing



Test of Independence

- Test of independence corresponds to an independent group of people who participate the testing process. They do not develop the software.
- 优势：1) 测试人员从中立的角度看待每个缺陷；2)测试人员完全没有偏见，即不带主观人至偏颇；3)测试人员对质量没有任何假设。
- 劣势：1) 与开发团队的隔离有时会导致过时的文档或版本引用；2) 独立的测试执行通常是最后一个阶段，在此过程中任何延迟都会影响受到版本或产品的发布；3) 开发人员可能对质量不负责任，因为他们可能认为独立测试团队的测试系统中有问题；4) 独立测试一定会受到通信阻碍。



Manual & Automatic Testing

人工测试	自动测试
消耗时间并单调：由于测试用例是由人力资源执行，所以非常缓慢并乏味。	时间消耗少：快速自动化运行测试用例时明显比人力资快。
人力资源上投资巨大：由于测试用例需要人工执行，所以在人工测试上需要更多的试验员。	人力资源投资较少：测试用例由自动工具执行，所以在自动测试中需要较少的试验员。



Manual & Automatic Testing

人工测试	自动测试
可信度较低：人工测试可信度较低是可能由于人工错误导致测试运行时不够精确。	可信度更高：自动化测试每次运行时精确地执行相同的操作。
非程式化：编写复杂并可以获取隐藏的信息的测试的话，这样的程序无法编写。	程式化：试验员可以编写复杂的测试来显示隐藏信息。



Internet Application System

❖ Internet Application System

- 中小规模企业：简单的Web页面
- 大规模企业：功能齐备的电子商务交易平台

❖ Characteristics in Structure

- The client-server (客户/服务器, C/S) structure. The client is the webpage, while the server is the web or the application server.



Testing the Internet System

❖ Testing Goal

- 在程序部署到Internet之前暴露其中存在的错误。消费者对Internet应用的质量期望较高，网站质量差可能会影响公司形象，甚至导致客户转向其他网站。

❖ Techniques

- 先前学习的测试技术同样适用于测试Internet应用系统，但测试Web应用系统时仍存在一些特殊之处。



Typical E-Business Structure

- Web Browser (网页浏览器)
 - 表现网站内容。虽然存在一定的标准，但不同厂商的产品中的增强功能导致浏览器运行不稳定，因此需测试兼容性。
- User Interface Layer (表示层)
 - 提供图形用户界面（GUI），Internet应用的外观和感觉来自本层，可以使用静态HTML或通用网关接口（CGI）脚本生成HTML。



Typical E-Business Structure

- Business Logic Layer (业务层)
 - 应用服务器对业务的逻辑处理过程，如事务处理、用户身份鉴定、数据确认、程序日志等。
- Data Access Layer (数据层)
 - 数据库服务器存储供应用系统使用或从最终用户收集来的数据，并基于数据库服务器对数据进行的各种操作（增、删、改、查）。数据源通常是关系数据库管理系统。



Challenges

- ❖ 基于Internet的应用系统中引起失效的地方很多，设计测试方法时必须考虑，例如：
 - 用户群：网站的用户群庞大且能力参差不齐，使用的浏览器、操作系统（OS）和设备种类不同，使用的信道速率差别也大。
 - 业务环境：网站须考虑财务和跟踪用户资料等问题。
 - 地点：用户可能位于其他国家，这时候涉及国际化问题，如语言、时差、货币兑换等问题。
 - 测试环境：测试时需要复制软件运行的环境，即使用与软件运行环境相同的Web服务器、应用服务器和数据库服务器，还要建立相同的网络环境。
 - 安全性：必须保证网站免受黑客攻击。

Examples

❖ 不同层有各自的特点，应将测试工作集中在特定的领域内，例如：

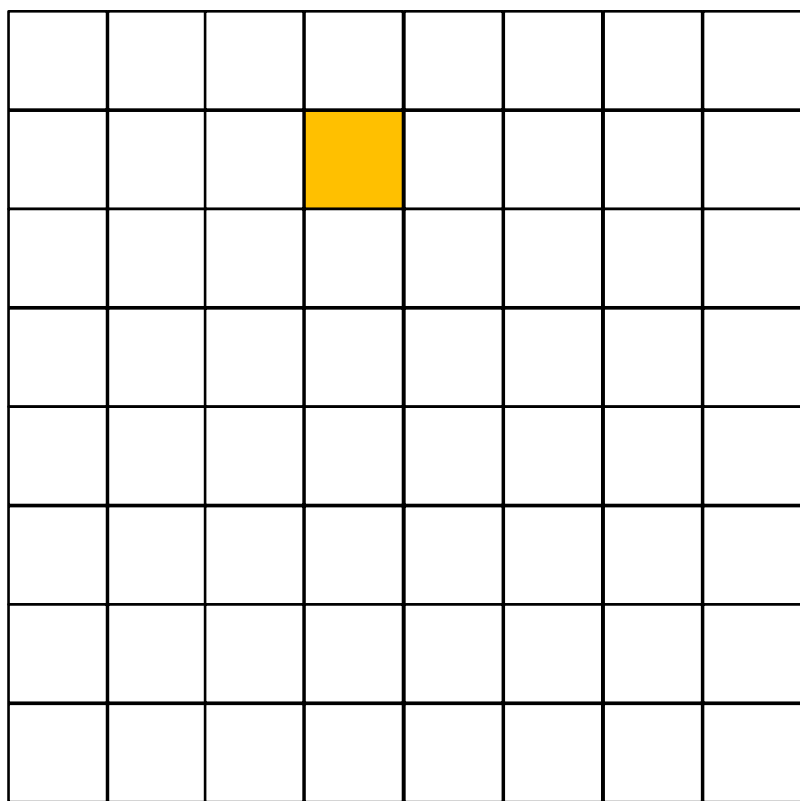
表示层	业务层	数据层
<ul style="list-style-type: none">• 确保字体在不同浏览器中都相同• 检查以确保每个链接都指向正确的文件或站点• 检查图形以确保其分辨率和大小正确• 对每页进行拼写检查• 让原稿编辑检查语法和风格• 在页面载入时检查光标位置，以确保其在正确的文本框中• 检查以确保在页面载入时选中了默认的按钮	<ul style="list-style-type: none">• 检查消费税和送货费计算是否正确• 确保提出的响应时间、吞吐率等性能指标得到了满足• 验证事务正确完成• 确保失败的事务正确回滚• 确保正确采集数据	<ul style="list-style-type: none">• 确保数据库操作满足性能要求• 验证数据存储适当且正确• 验证可使用当前备份来恢复• 测试故障处理和冗余功能



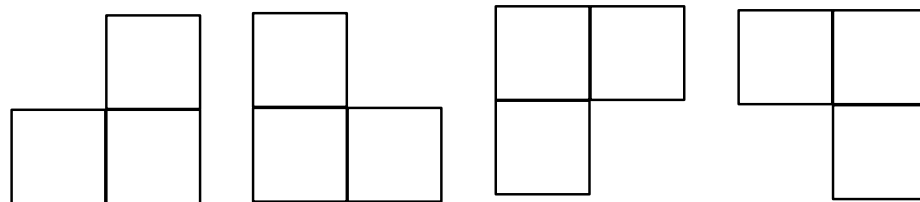
Strategies

- ❖ 测试基于Internet的应用系统需要深入了解组成应用系统的每个硬件和软件组件，还需要一份规格说明文档来描述Web站点的预期功能和性能：
 - 内部开发的部件：验证其符合设计规格说明、需求规格说明文档中描述的功能要求后方可将其集成到系统中；可使用单元测试、代码审查等技术。
 - 第三方部件：设计系统测试以确认这些部件可以独立于应用正确执行，接受后方可集成。
- ❖ Adopting the divide-and-conquer method (分治法)
 - Each layer has its own characteristics, so it is better to use the divide-and-conquer method.

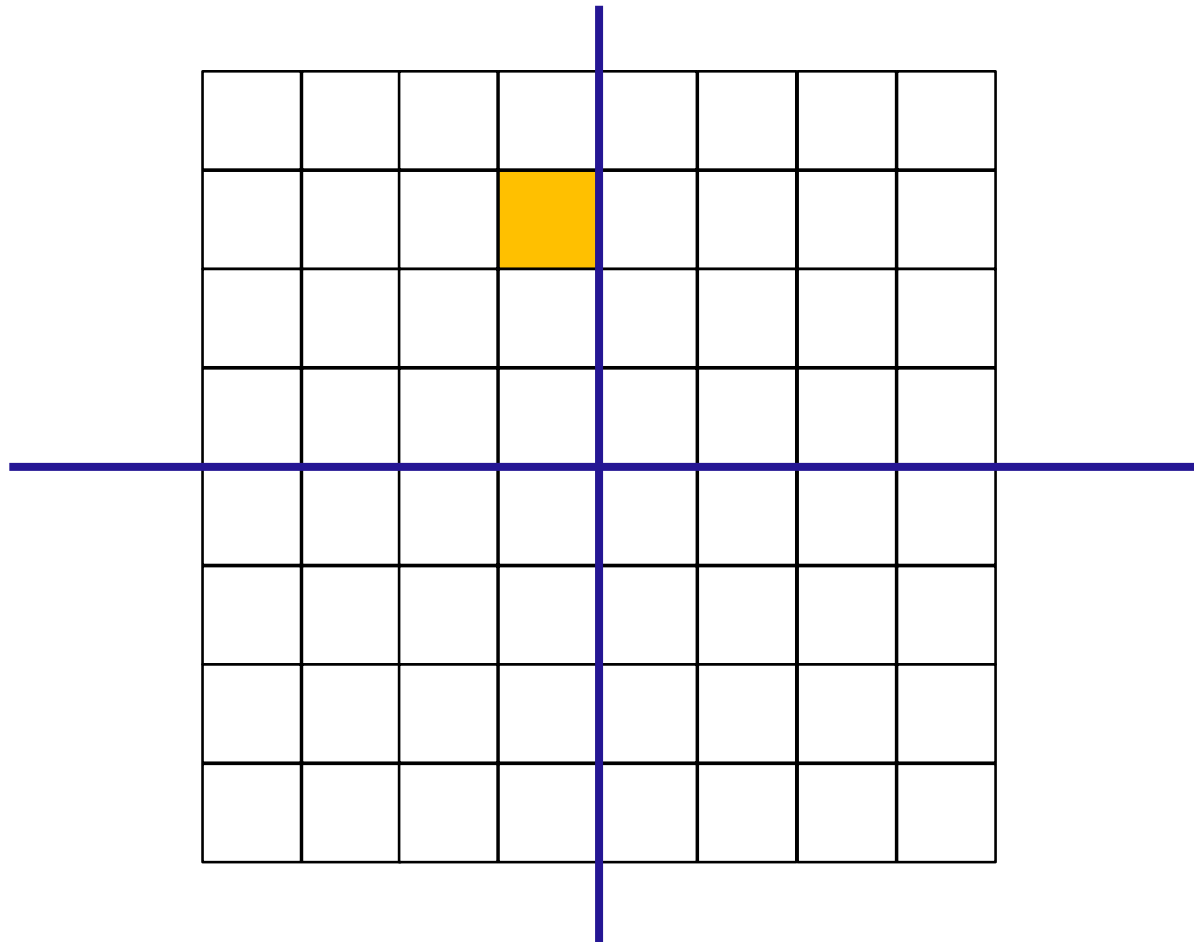
Divide-and-Conquer Method



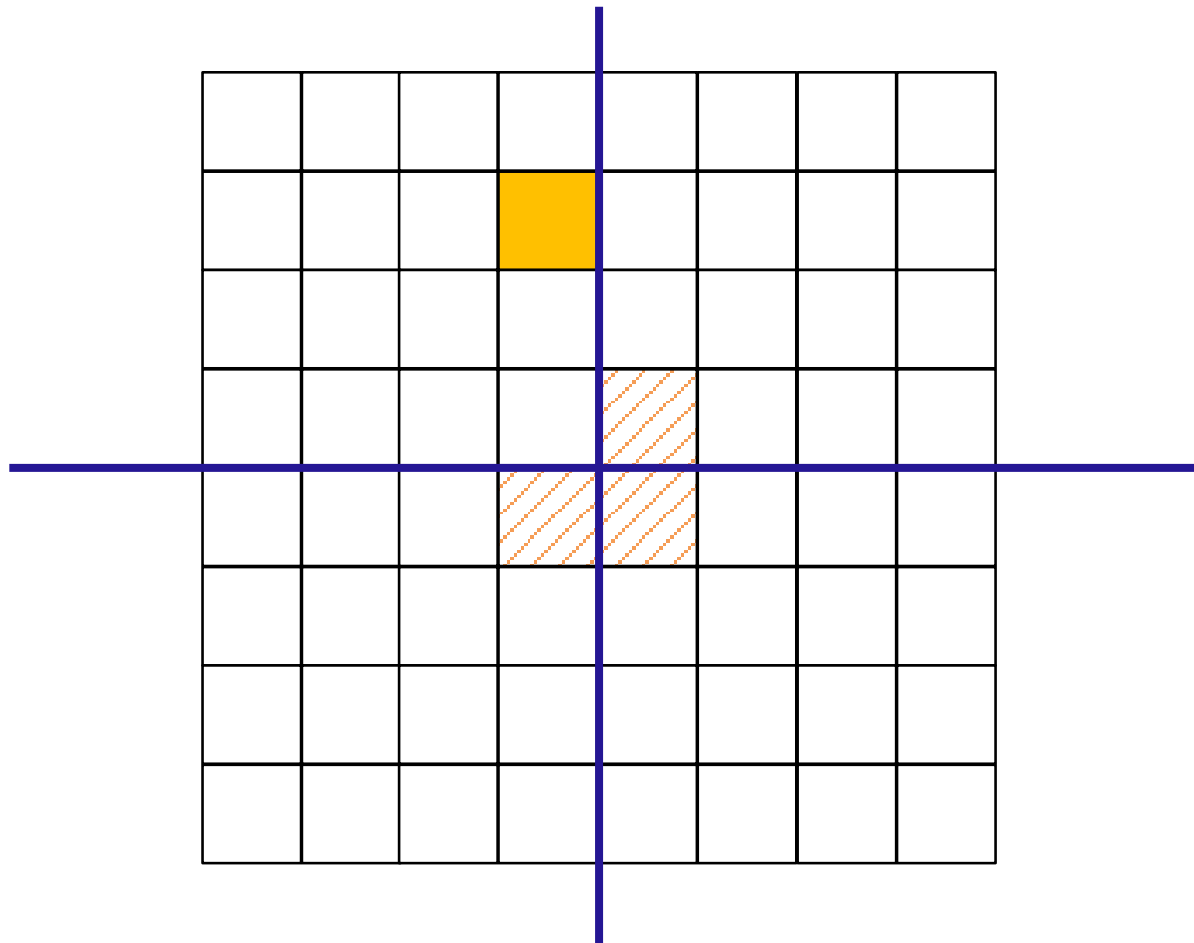
- 如何使用下列L型骨牌
不重叠地恰好覆盖除
了涂色方块外的整个
 8×8 棋盘？



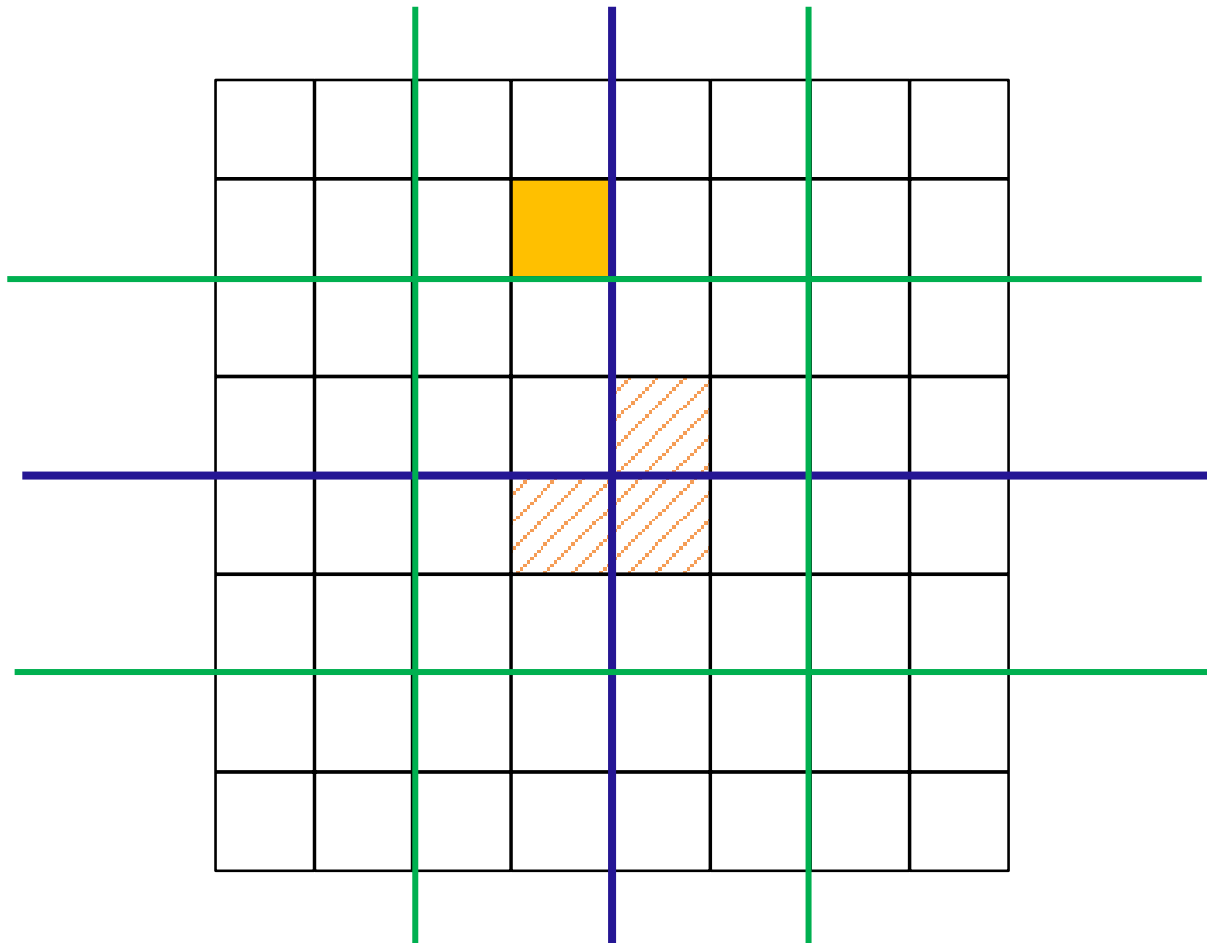
Divide-and-Conquer Method



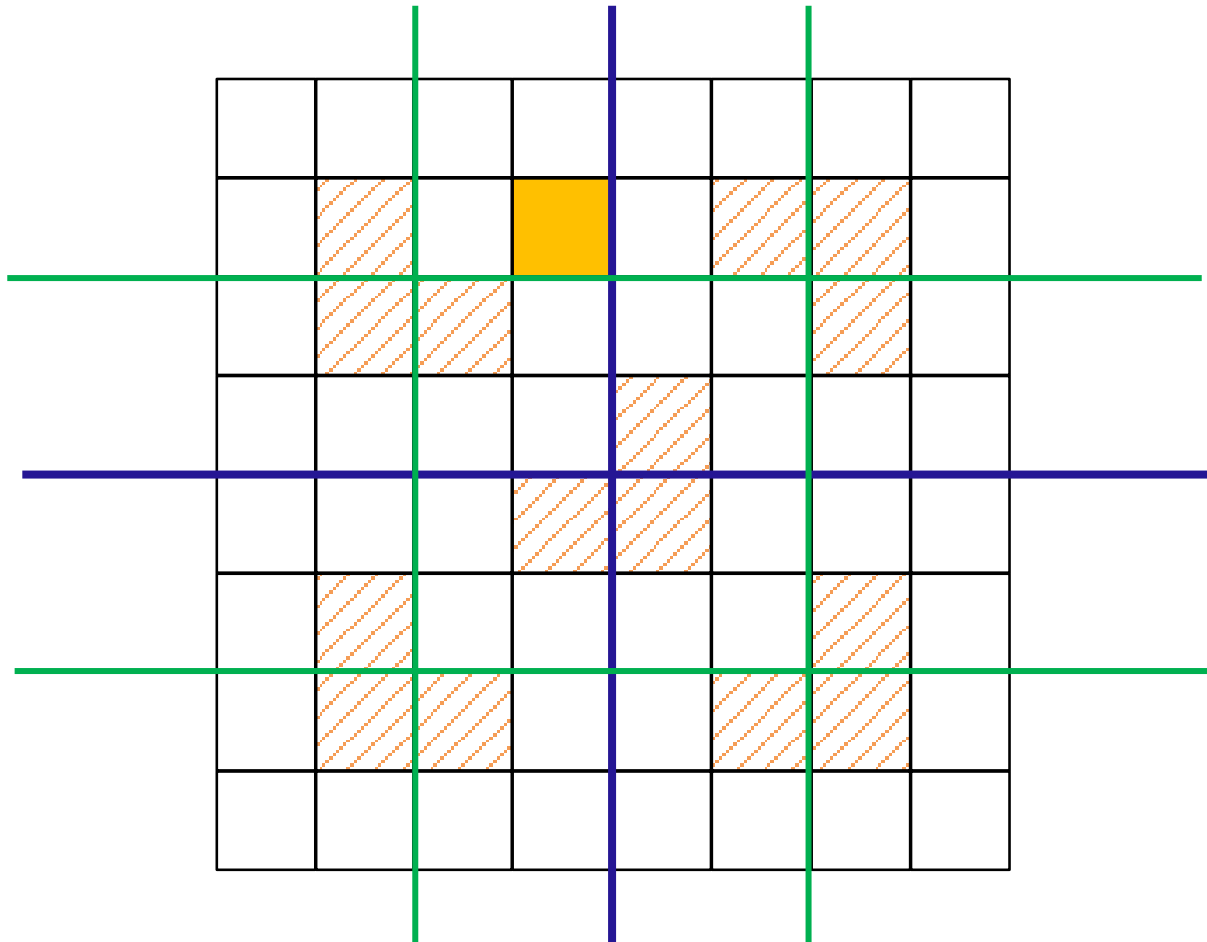
Divide-and-Conquer Method



Divide-and-Conquer Method



Divide-and-Conquer Method





Testing the Three Layers

- Testing the User Interface Layer
- Testing the Business Logic Layer
- Testing the Data Access Layer




Testing the Three Layers

- Testing the User Interface Layer
- Testing the Business Logic Layer
- Testing the Data Access Layer



Testing the UI Layer

- ❖ 测试目的：发现应用程序的GUI或前端中的错误
- ❖ 表示层测试的主要内容
 - 内容测试
 - Web站点结构测试
 - 用户环境测试



Content Testing

- 内容测试：整体审美、字体、色彩、拼写、内容准确性和默认值。
 - 检查人机界面元素，在字体类型、屏幕布局、色彩、图形分辨率及其他会直接影响用户体验的特性中检查错误。
 - 检查Web站点中信息的准确性，包括语法和内容。



Web Structure Testing

- Web站点结构测试：无效的链接或图形。
 - 尽量发现浏览过程和结构上存在的错误，发现无效的链接、丢失的网页、错误的文件或其他任何将用户引到站点中错误区域的问题。
 - 有很多工具可以自动执行验证链接、检查丢失文件等过程。
 - 白盒测试可以用于测试Web页面。



User Environment Testing

- 用户环境测试：Web浏览器版本和操作系统配置。
 - 测试不同浏览器、同一浏览器的不同版本。
 - 如果使用ActiveX控件、JavaScript、VBScript、Java applet时应该特别关注浏览器的兼容问题。



Testing the Three Layers

- Testing the User Interface Layer
- **Testing the Business Logic Layer**
- Testing the Data Access Layer



Testing the Business Logic Layer

- ❖ 测试目的：发现Internet应用系统的业务逻辑中的错误
- ❖ 测试技术：类似于测试单机程序，可以使用黑盒测试和白盒测试技术
 - 黑盒测试是业务层测试的主要方法：模拟用户使用过程
 - 可以使用白盒测试测试内部开发的部件
- ❖ 测试内容
 - 性能测试：检查应用系统是否满足书面的性能规格说明
 - 数据验证测试：发现从客户那里采集到的数据中的错误
 - 事务测试：发现事务处理过程中的错误



Performance Testing

❖ 性能测试

- 长时间页面加载、缓慢的事务处理会使用户怀疑其鲁棒性。这样的强度测试通过大量的并发操作使系统临近失效点，借以判断应用系统是否满足其性能目标，还可以检查网络设施的鲁棒性。
- 性能规格说明应该在需求阶段完成，通常以响应时间或吞吐率描述。



Data Verification Testing

❖ 数据验证测试

- 确保收集来的数据是有效的，若使用无效信息可能导致严重错误。
- 与测试单机应用时查找用户输入或参数的错误很相似，应通过测试发现数据采集时的错误。



Transaction Testing

❖ 事务测试

- 电子商务网站必须在全部的时间里正确处理事务。可以将事务测试看作是对业务层的系统测试。
- 除了处理测试内部的业务过程之外，还必须测试外部服务（第三方服务）。



Testing the Three Layers

- Testing the User Interface Layer
- Testing the Business Logic Layer
- Testing the Data Access Layer



Testing the Data Access Layer

❖ 数据层测试的特点

- 主要对用于储存和获取信息的数据库管理系统的测试。
- 数据测试的最大挑战之一是复制应用系统的运行环境，必须使用相同的硬件平台和软件版本来进行有效的测试。



Testing the Data Access Layer

❖ 测试数据层时应当在特定的方面查找错误

- 响应时间测试
- 数据完整性测试
- 容错性和可恢复性测试



Response Time Testing

❖ 响应时间

- 测量响应时间，确定每一项数据库操作（插入、删除、修改、查询）需要多久完成；事务的完成时间涉及多次数据库操作。
- 大多使用黑盒测试。



Data Integrity Testing

❖ 数据完整性测试

- 数据完整性指的是数据存储适当并且正确，因此测试就是在数据库表中发现不准确数据的过程。
- 尽力在数据的存储方式中发现问题，如数据类型和长度导致数据失去精度的问题；还要检查系统使用的参考数据的准确性，如邮政编码、时区等信息。



Fault-Tolerant & Restorability Testing

❖ 容错性和可恢复性

- 最大化平均故障间隔时间（MTBF）：取决于数据库的容错级别：可能通过故障处理机制或内置的容错机制，使得数据库故障时对系统的影响最小，根据系统采用的结构来决定采用的测试类型。
- 最小化平均恢复时间（MTTR）：可恢复性测试的目标是设计出数据库无法恢复的场景，恢复计划开始于获得有效的备份，测试时若无法恢复数据库，就需要修改备份策略。



Webpage Testing

- 网页中既有黑盒测试又有白盒测试。黑盒测试的内容包括了文本、超链接、图片、表单、对象，以及一些比较简单的功能（如滚动文本选择框）。
- 白盒测试需要在了解网站的系统结构和编程知识的基础上进行，这些基础内容包括动态内容、数据库驱动的网页、编程方式创建的网页、服务器性能和加载、安全性能等。
- 此外，还有一种介于黑盒和白盒之间的灰盒测试，即以黑盒测试为主，辅助以简单查看内部工作机理作为补充。



THANK YOU!