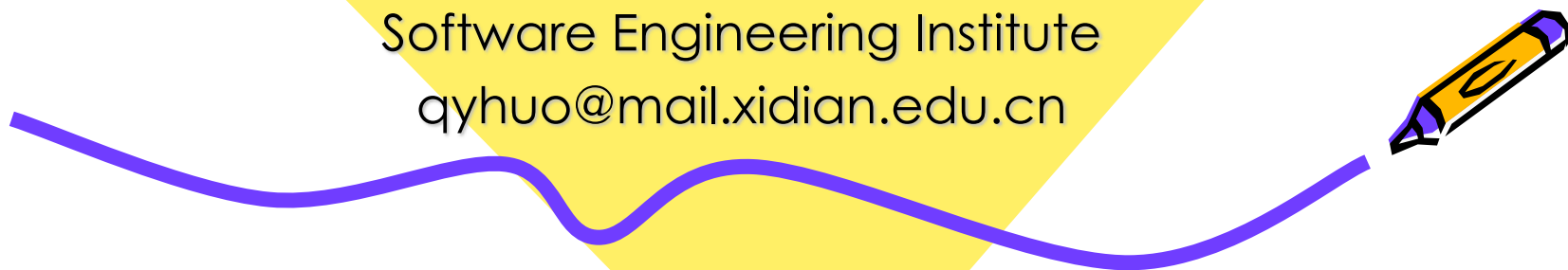




Web应用测试

Qiuyan Huo 霍秋艳
Software Engineering Institute
qyhuo@mail.xidian.edu.cn

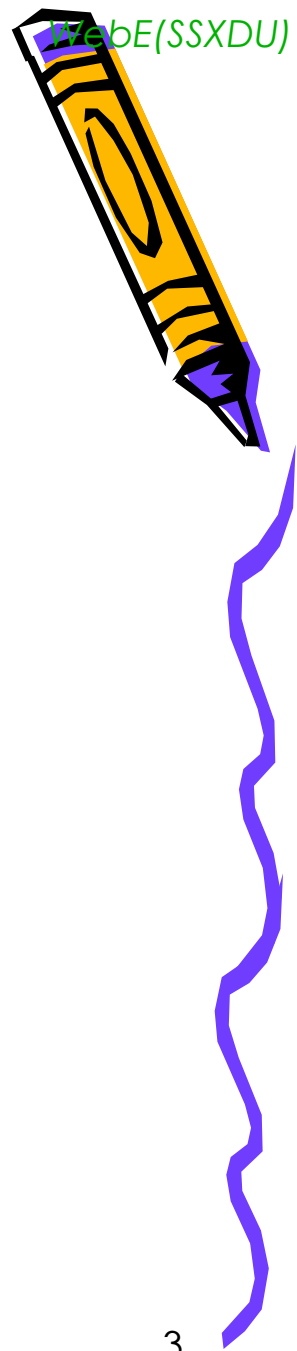


Web Applications have been evolved
and adopted as essential
communication and business
platforms for many companies and
their customers.

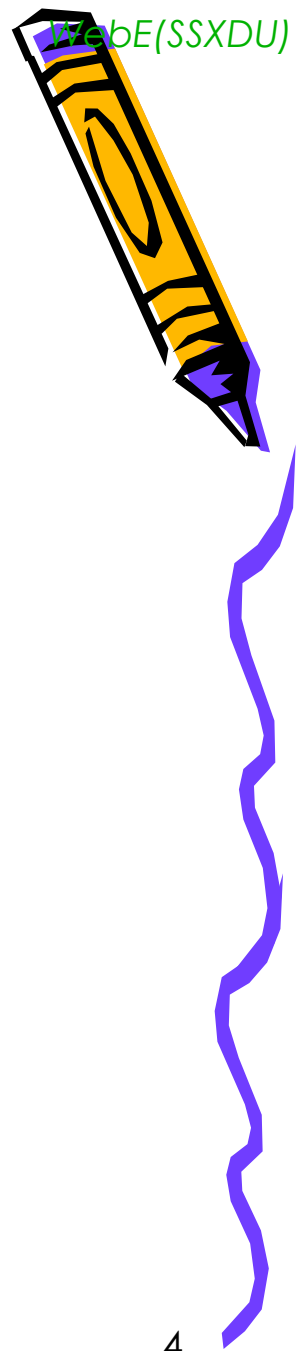


Web应用测试

- Web应用测试特性
- Web应用测试过程
- 功能测试
- 内容测试
- Web页面测试
- 兼容性测试
- 性能测试
- 安全性测试
- 接口测试
- Web服务测试
- 测试工具
- 总结与展望



WEB应用测试特性





<head>

<!-- CQVM:p002 CQRENDERTIME:Fri, 07 Jun 2013 11:09:01 -05:00 -->



Hello! Sign Up

My Account | Feedback | Customer Support or call 1.888.872.8356

GO

Home

Vacation Packages

Flights

Hotels

Cars/Rail

Cruises

Travel Deals

Activities

Las Vegas



1. Select an option to start your travel search

There are errors in the form below.

- ☐ SAVE! Flight + Hotel
- ☐ Hotel + Car

- ☐ Hotel Only
- ☐ Flight + Hotel + Car
- ☒ Flight Only
- ☐ Car Only
- ☐ Cruise

2. Enter your origin and destination cities

Form Navigation

- ☒ Round-Trip
- ☐ One-Way

From: To: [Multiple](#)

[Destinations?](#)

☐ Compare Surrounding Airports

3. Choose your travel dates

- ☒ Exact Dates
- ☐ +/- 1 to 3 Days

US/Canada Only

- Search Your Choice of Airports for the Lowest Prices.
- Compare the Best Prices on and Around Your Dates.
- Find the Best Days to Travel.

Depart Date: Depart Time: Flexible Days:

mm/dd/yyyy

Anytime +/- 3 Days

**Save up to 40%
on Hawaii Hotels**
Book Your Vacation Today!



The
HAWAIIAN
ISLANDS

Book Now

[-]

Feedback

Web pages became very complex.

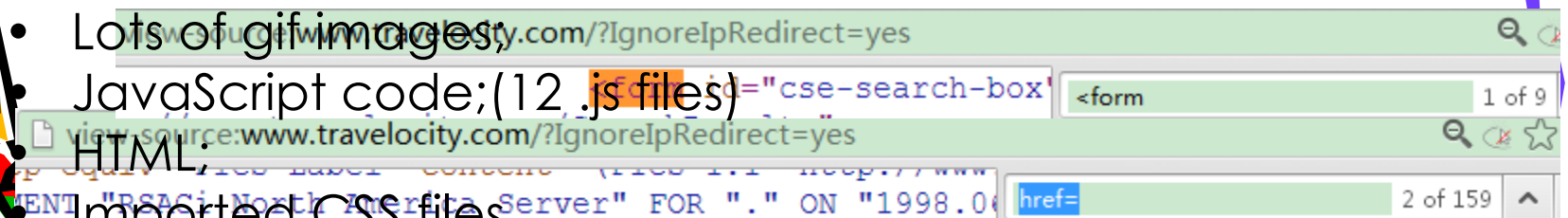
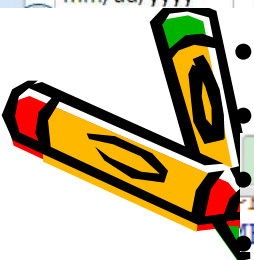
• Over 100 different actions (over 150 links and 9 forms);

• Lots of gif images;

• JavaScript code;(12 .js files)

• HTML;

• Imported CSS files



Many Things Can Go Wrong

- Interactions between HTML pages (broken or orphan links)
- Applications that run in Web browsers (e.g., applets, JavaScript, plug-in applications)
- Applications that run on the server side (e.g., cgi scripts, business rules, integration with other apps, database connections, logging applications, dynamic page generators, asp, jsp)
- Wide variety of servers and browsers (heterogeneous systems)
- Rapidly changing technologies, multiple standards and protocols (not easy to follow)



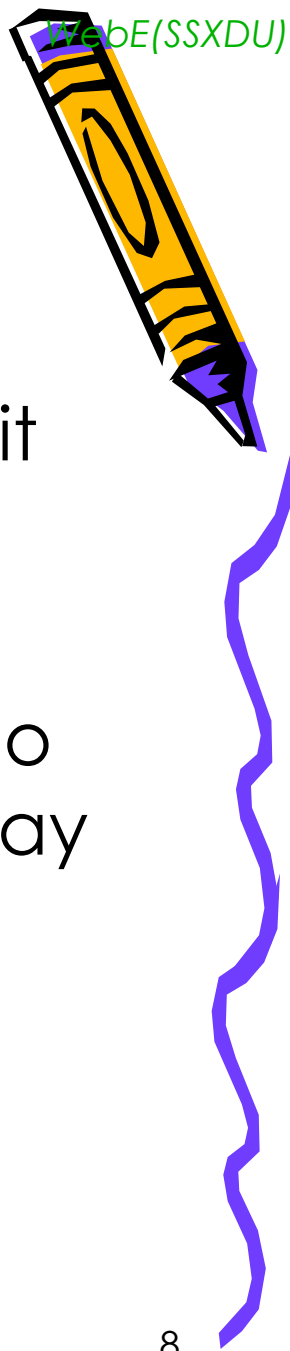
Many Things Can Go Wrong, cont.

- Inconsistency software and non-convergence technologies
- Information security (vulnerability)
- Networking issues (latency, protocols, etc.)
- Legacy systems (data inconsistency, missing transactions)
- Performance (architecture incompetence)
- Maintenance and supportability
- Legal and privacy issues (outdated)
- Usability (hard to find useful information or instructions)
- Scalability (not scale as the volume increases)



What to do?

- Properly and symmetrically test the system and control the quality before it goes wild.
- The only cause is our human being who makes errors all the time. There is no way to get around.



Terminology

- Some definitions:
 - **Testing**: An activity conducted to evaluate the quality of a product to improve it by identifying defects and problems.
 - **Error**: the actual result deviates from the expected.
 - Our expected results *should* (theoretically) come from our requirements definition.
 - Most often, the goals/concerns/expectations of stakeholders serve as the testing basis.
 - **Test case**: a set of inputs, execution conditions, and expected results for testing an object.

Test Objectives

- *Main objective*: find errors, especially critical errors.
- Complete test coverage(覆盖) is **impossible**, so testing focuses on **mitigating**(降低) the largest risks.
 - Where's the greatest potential for loss?
 - What are the sources of this risk?
- Start testing as early as possible – even with restricted resources and time.



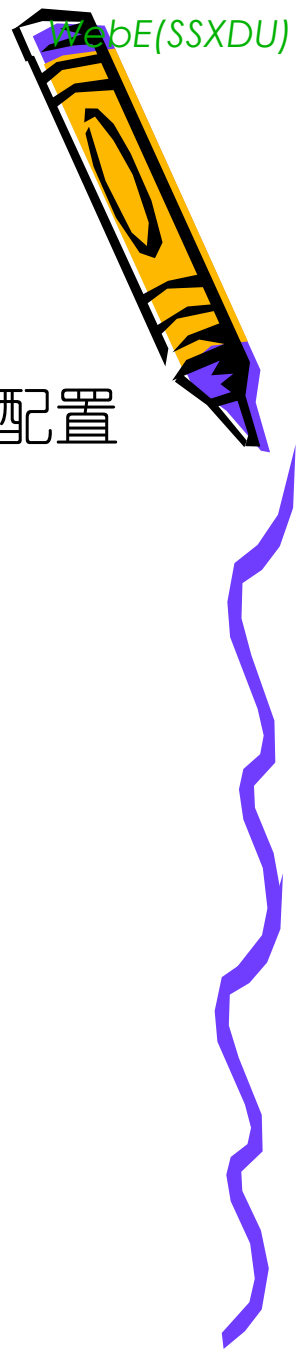
Web应用测试特性

- 用户量大、资源的跨平台全局访问、处理并发事务
 - 需要进行多用户访问的性能测试
- “内容”中的错误、超文本结构中的页面之间链接关系、展示层的“软”需求（如艺术性）、全球性
 - 使得诸多方面测试难度大
- 多层架构
 - 确定系统功能或性能缺陷的位置难度大

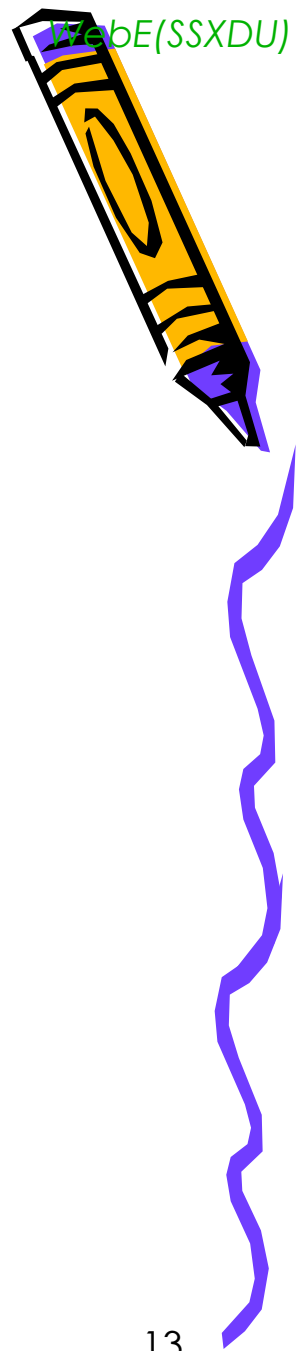


Web应用测试特性

- 集成特性
 - 测试必须包括对第三方软件和组件及其集成和配置的测试
- Web浏览器提供的导航，如回退（back）
 - 经常引发各种错误
- 技术种类
 - 定义测试粒度和建立组件间的关系困难
- 运行环境异构、自治
 - 要求针对其特性分别选择测试方法

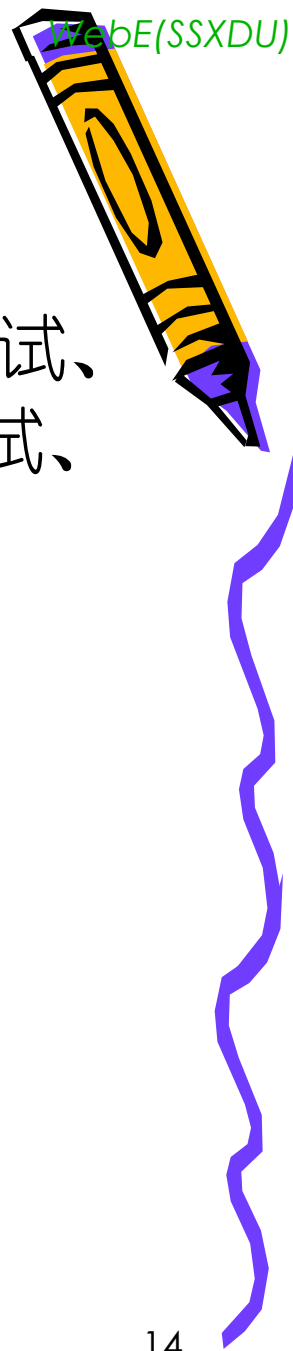


WEB应用测试过程



Web应用测试一般过程

- 功能测试、内容测试和评审、Web页面测试、导航测试、接口测试、配置测试、安全测试、性能测试。



Web应用测试内容

- 对被测试的Web应用进行需求分析

测试的目标和范围，所测试的目标需要实现一个什么样的功能，总结基本文档、主要活动

- 定义测试第

测试开始的条件、测试的类型、测试开始的标准以及所测试的功能、测试通过或失败的标准 制定测试的条件和处理

- 确定测试环

选择合适的测试用例、选择合适的测试工具、决定执行测试的人员，以及确定测试要做到何种程度（测试的充分性）。

- 描述测试的

测试用例列表，进度表，错误等级分析，对测试计划的总结，测试过程会出现的风险分析等。

Web应用测试过程

- Web应用测试任务

Web应用测试

1. 评审利益相关者的需求。
 - 标识关键用户目标。
 - 对每类用户的用例进行评审。
2. 建立优先级，以确保每一个用户目标都将被适当地测试。
3. 根据要实施的测试类型的描述定义Web应用测试策略。
4. 制定测试计划。
 - 规定测试进度，并对每个测试分配职责。
 - 指定自动化测试工具。
 - 规定每一类测试的验收标准。
 - 详细说明缺陷跟踪机制。
 - 定义问题报告机制。
5. 进行“单元”测试。
 - 评审内容的语法和语义错误。
 - 评审内容的许可性。
 - 测试接口机制的正确操作。
 - 测试每一个组件（比如，脚本），确保正确的功能。
6. 进行“集成”测试。
 - 对照用例来测试界面的语义。
 - 实施导航测试。
7. 进行配置测试。
 - 评估客户端的配置兼容性。
 - 评估服务器端的配置。
8. 进行性能测试。
9. 进行安全性测试。

Agile Approaches

- Rely on self-organization
- Testing is an integrated development activity.
 - Remember, tests are developed before code.
 - Everyone in the team is responsible.
- Unit tests are automated, acting as “change detectors”
 - Immediate feedback
 - Lends itself to short development cycles.



The agile approach described here mainly to unit and acceptance tests.

Test Levels

- Developers are not testers
- Users are not testers
- They do not share test cases



Example – Unit Test (in Java)



```

@Test (groups = {"functional", "negative"},
expectedExceptions=ConstraintViolationException.class)
public void addExistingAccount()
{
    Account account = new Account();
    account.setFirstname("Jay");
    account.setLastname("Smith");
    account.setEmail("jsmith@example.com");
    account.setUsername("jsmith");

    accountDao.add(account);
}

```

• Unit test expression

- Do not rely on human judgment
- Easy to run many at the same time



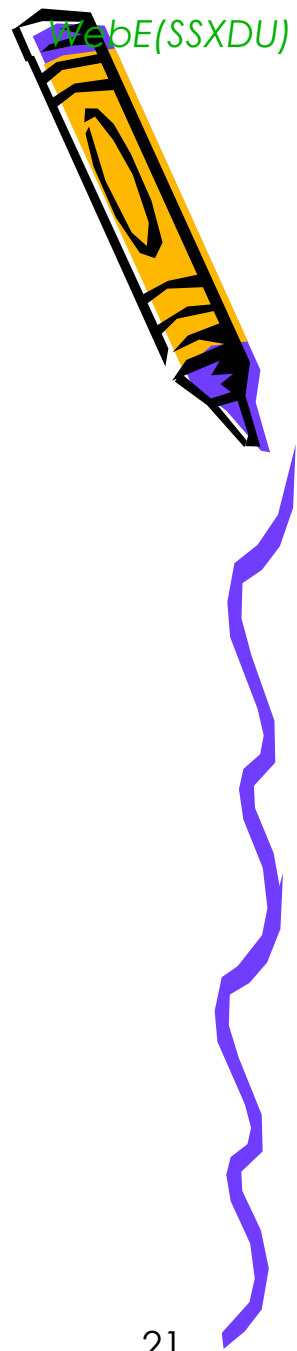
Your Role as a Tester

- The ideal tester has a “destructive” attitude.
- Very difficult for developers to “destroy” their own work.
- Thus, some guidelines:
 - Have others in the Web team generate test cases and perform unit tests.
 - Test cases are generated from business requirement documents, not from programs.
 - Best tester is the one who gets the most bugs **fixed**.

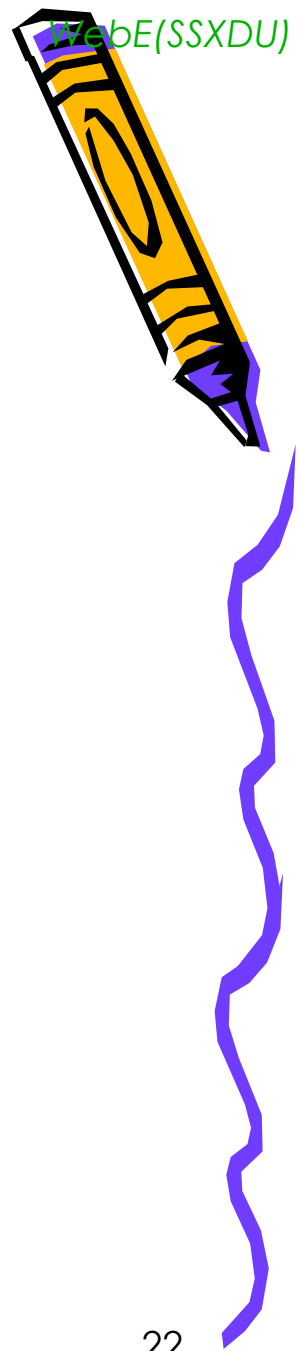


How to handle defects

- Defect reporters
 - Testers
 - Dedicated teams(专责团队)
 - Users
- Defect classified by
 - PM (project managers)
- Defect fixed by
 - Developers

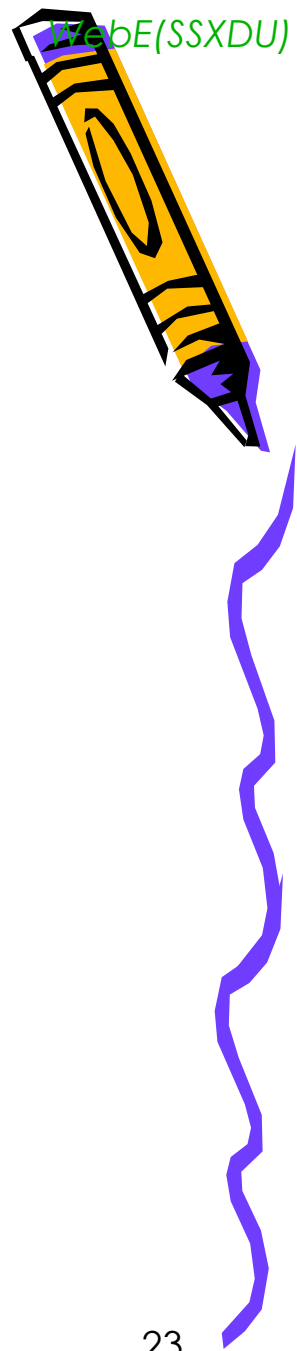


功能测试



功能测试

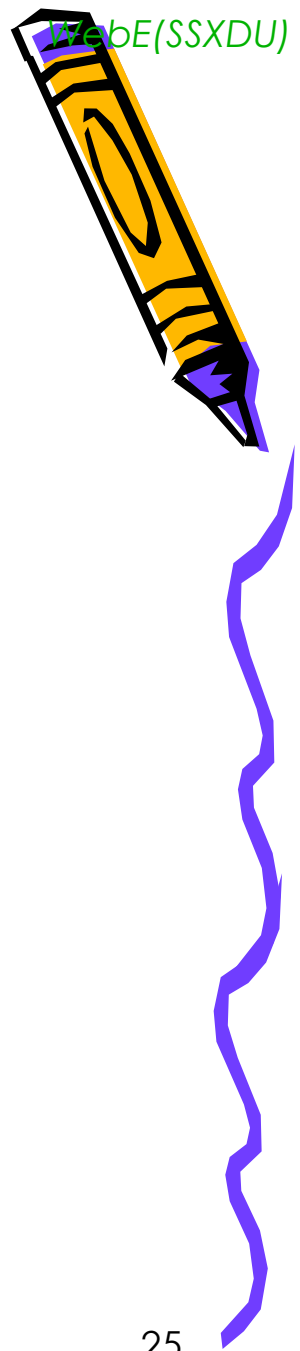
- 链接测试
- 交互测试
- 数据校验
- Cookies测试
- 数据库测试
- 特定功能需求测试



Link Testing

- Finding broken links
 - Can be automated through a spider
 - Doesn't help for pages with no incoming links.
- Finding orphan pages
 - Orphans are pages with no links back to the navigation structure.
 - Users get frustrated and leave.
- Capturing statistics
 - Depth & breadth of navigation.
 - Distance between two related pages.
 - # of links.
 - Load times.





内容测试



内容测试的目标

- 找出基于文本的文档、图形展示和其它媒体中的语法错误，例如，打字错误、语法错误；
- 找出当导航发生时所展现的任何内容对象中的语义错误；
- 找出展示给最终用户的内容的组织或结构方面的错误。

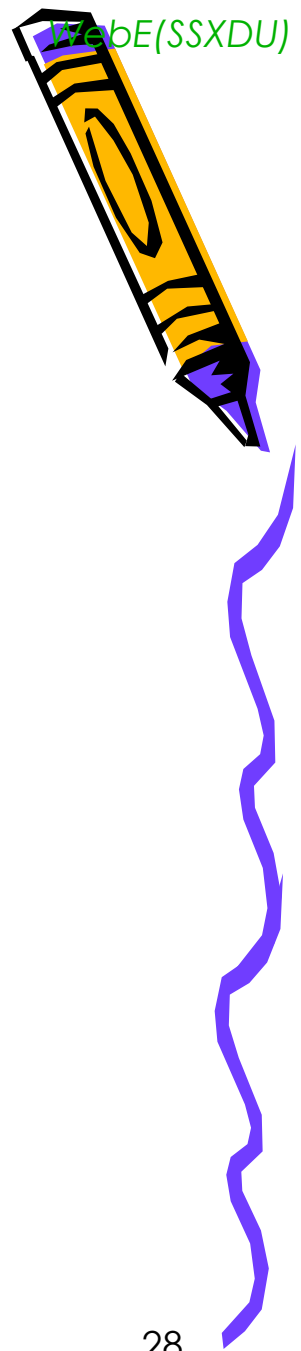


验证动态内容

- 对用户界面层进行测试，确保对每一个用户查询都正确地构造了Web页面脚本，并且正确地传输给了服务器端；
- 对服务器端的Web应用层进行测试，确保能够从Web页面脚本中正确地抽取出用户数据，并且正确地传输给服务器端的数据转换层；
- 对数据转换功能进行测试，确保创建了正确的SQL，并且传给合适的数据库管理组件。

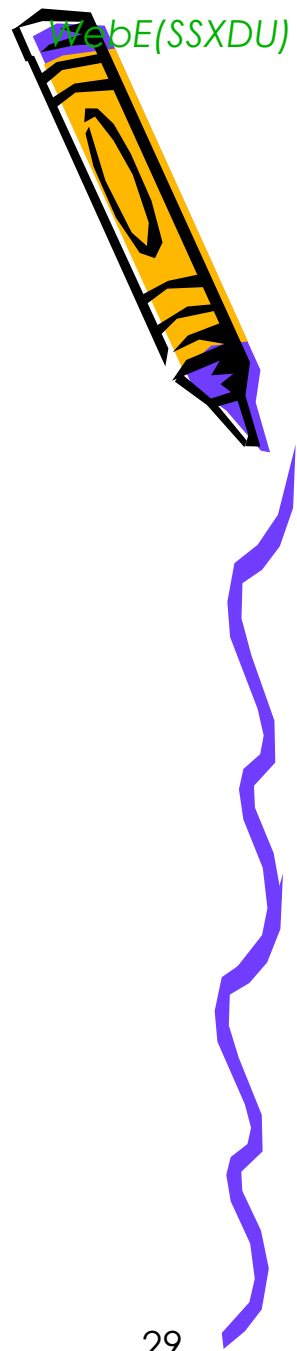


WEB页面测试



Web页面测试

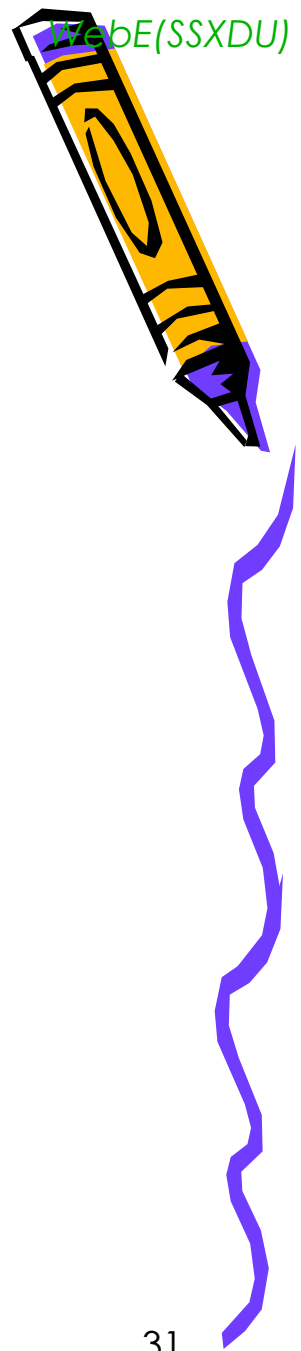
- Web页面测试内容
- 导航测试
- 图形测试
- 内容测试
- 表格测试
- 整体界面测试



Web页面测试的要素

要素类型	面对的问题
说明和技术信息	信息和指令的准确性
字体	样式一致性 字体清晰度 识别斜体和截线字体的困难度 一篇文档中多种字体造成的视觉混淆以及目标平台上字体的可用性
颜色	背景颜色的适宜性 前景颜色的适宜性 字体颜色的适宜性 精细、互补颜色的选择通常比饱和、反差色更悦目
边缘	命令按钮的三维效果对用户是有效的可视化提示 对非交互元素三维效果的使用会被混淆
图像	大图像可能增加装载时间 可视化提示和设计细节同背景是否能区分 背景的适宜性 标签清晰度 按钮清晰度 图片尺寸的适宜性
框架	浏览器能否正确显示 显示设置和浏览器类型影响框架的显示情况 回退按钮经常有意想不到的结果
表格	网格（表格中的表格）减慢HTML的装载速度 外观是否由于显示设置和浏览器类型而导致不正确的范围和重叠 测试应该包括所有浏览器、显示设置和浏览器窗口大小

兼容性测试



Compatibility(兼容性) testing

- Determines if an application runs as expected given a running environment that has various combinations of hardware, software, and middleware.(确定给定的环境下是否按照期望运行)
- Often the large variety of possible combinations of all the components part of the execution of a WebApp does not make it feasible to test them all.(一个WebApp执行的所有组件的可能组合量太大)
- Usually only the most common combinations are checked. As a consequence, just a subset of possible compatibility failures might be discovered.(通常只测到最可能的组合——部分)



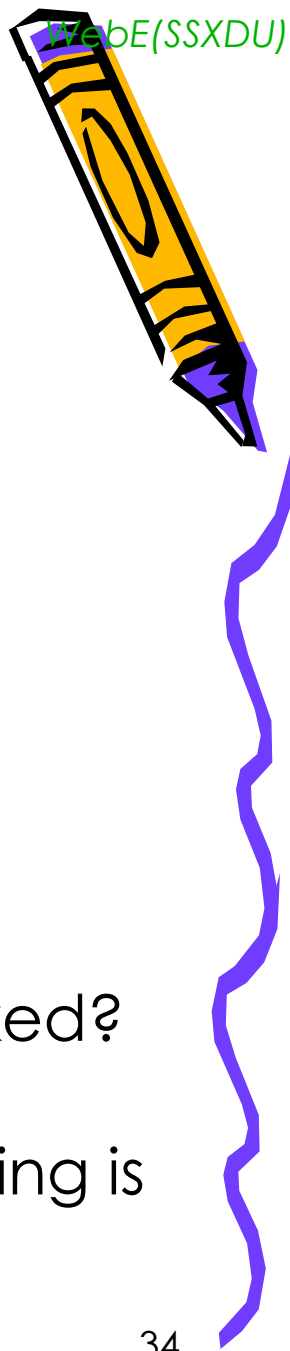
兼容性测试

- 平台测试
- 浏览器测试
- 分辨率测试
- 连接速率测试
- 打印机测试
- 数据库兼容性测试
- 应用软件间兼容性测试

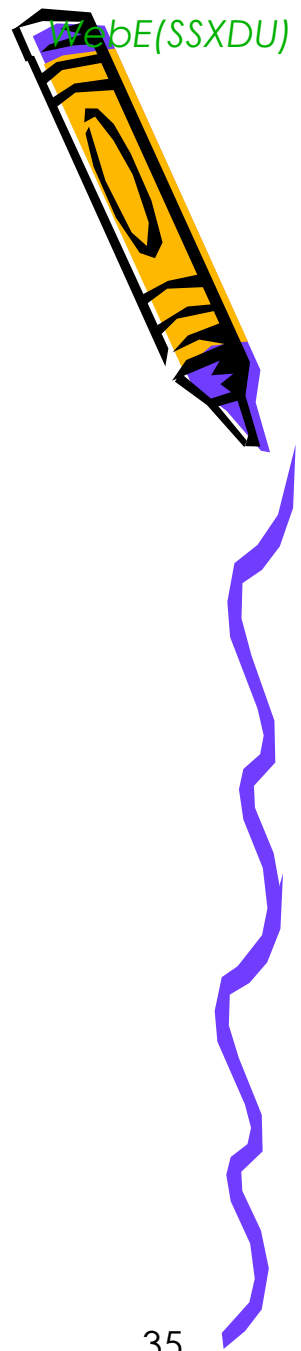


Browser Testing

- Browsers vary by:
 - Manufacturer
 - Version
 - Operating system
 - Device
 - Configuration (stylesheets, JavaScript on/off)
 - W3C Standard compliance
- Important questions to ask:
 - How is state managed?
 - Can a (dynamic) Web page be bookmarked?
 - Can users open multiple windows?
 - What happens when cookies and/or scripting is turned off?

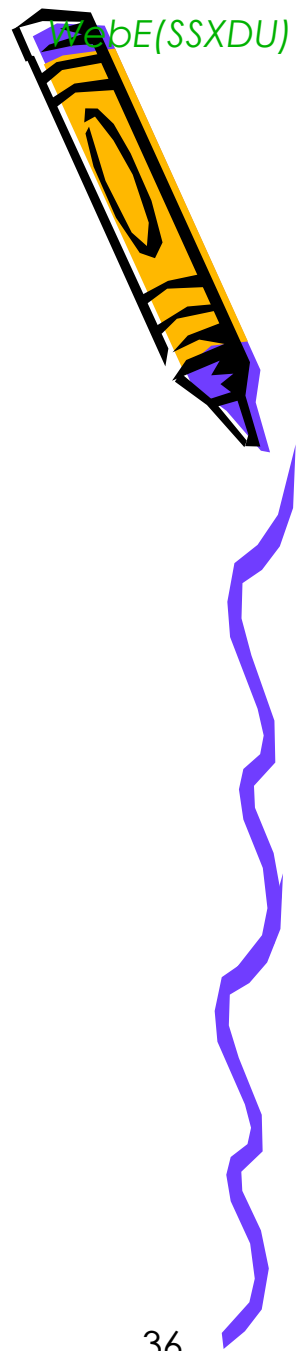


性能测试

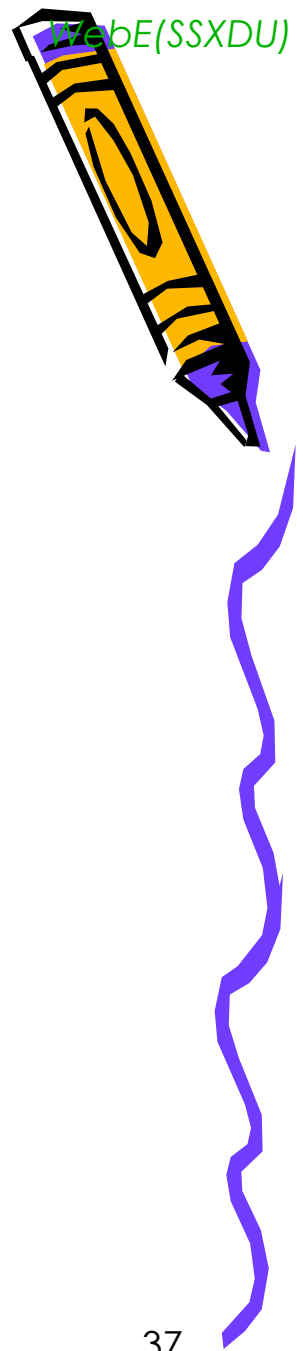
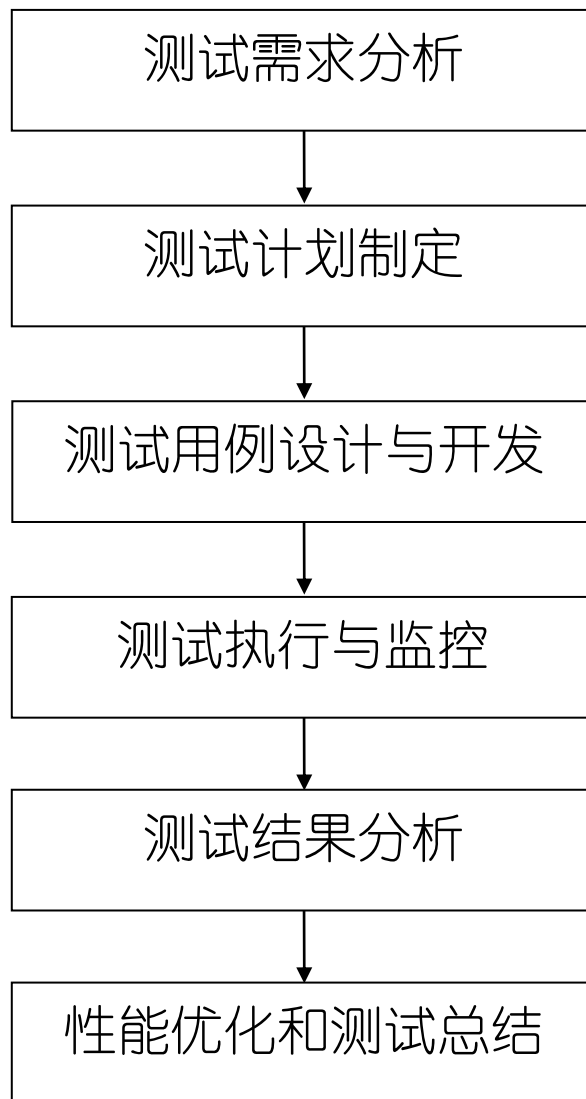


性能测试目标

- 评估系统性能
- 确定系统瓶颈
- 系统调优
- 检测Web应用中潜在的问题

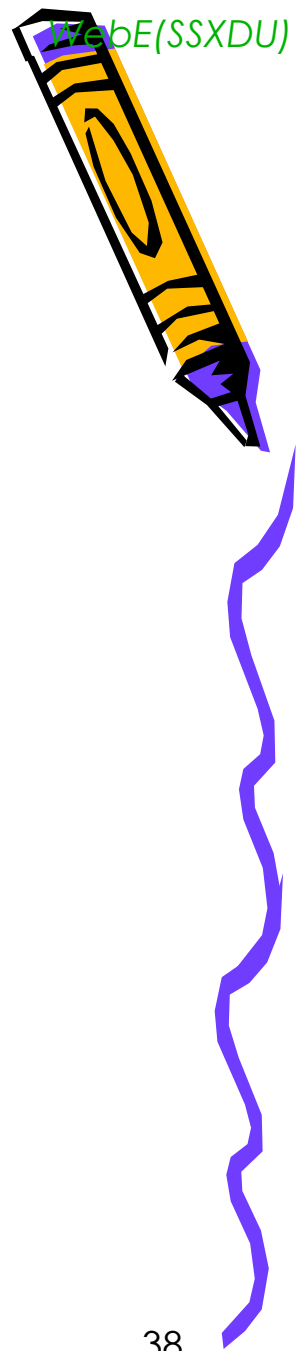


Web应用性能测试过程



性能测试内容

- 速度测试
- 负载测试
- 压力测试
- 并发测试
- 大数据量测试
- 配置测试
- 可访问性
- 可靠性测试



Performance testing

- verifies specified system **performance factors** (e.g. response time, service availability).
 - by **simulating** hundreds or more simultaneous user accesses over a defined interval.
- Information about accesses is recorded and then analyzed to estimate the **load levels** exhausting the system resources.
- Failures that can be uncovered
 - by performance testing are mainly due to running environment faults (e.g. scarce resources, poorly deployed resources)



Load Testing

- Does the system meet required **response times** and **throughput**?
- **Load profile** - expected access types, visits per day, transaction types, transactions per session, etc.
- Must determine the **range** of values for response times and throughput.
- **Evaluate** the results to look for bottlenecks.



Load Testing

- To determine how the WebApp and its server-side environment will respond to various **loading conditions**
 - N, the number of concurrent users
 - T, the number of on-line transactions per unit of time
 - D, the data load processed by the server per transaction
- Overall throughput, P, is computed in the following manner:
 - $P = N \times T \times D$

Stress Testing

- How does the system behave under **abnormal/extreme** conditions?
- The test should tell you...
 - If the system meets the target responses times and throughputs
 - If the system responds with an appropriate error message. (i.e. graceful degradation)
 - If the system crashes (it should NOT!)
 - How quickly the system recovers to normal operation.

LoadRunner..., WAS...

Accessibility (可访问性) testing

Web内容对于各类人员、各种访问设备（如桌面浏览器、语音浏览器、移动电话、车载个人电脑等）或者在条件限制的情况下使用。

- 保证内容良好呈现
- 使内容易理解与可导航
- TAW (Web Accessibility Test) is a tool for the analysis of Web sites, based on the [W3C - Web Content Accessibility Guidelines 1.0](#) (WCAG 1.0)

Usability Testing

- What are the 2 main approaches to usability testing (engineering)?
- We've also talked about accessibility and evaluation approaches
 - Visual, cognitive, physical, and age-related should garner(得到) the most attention when testing.
 - <http://validator.w3.org>



Usability Tests

Bad usability

- Costs time and money
- Is a safety risk
- Is a competitive disadvantage
- Lowers customer/employee satisfaction



Usability testing

- Design by WebE team ... executed by end-users
- Testing sequence ...
 - Define a set of usability testing **categories** and identify **goals** for each.
 - Design tests that will **enable** each goal to be evaluated.
 - Select **participants** who will conduct the tests.
 - Instrument participants' interaction with the WebApp while testing is conducted.
 - Develop a **mechanism** for assessing the usability of the WebApp
- different levels of abstraction:
 - the usability of a specific interface mechanism (e.g., a form) can be assessed
 - the usability of a complete Web page (encompassing interface mechanisms, data objects and related functions) can be evaluated
 - the usability of the complete WebApp can be considered.

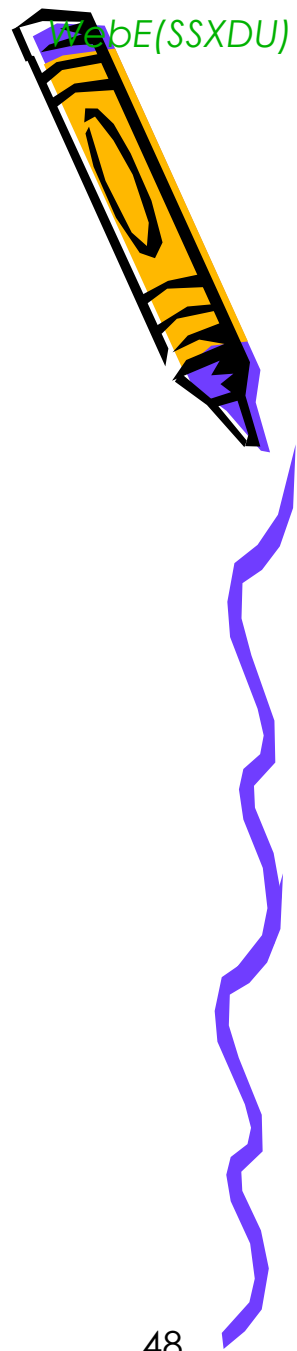
Usability testing

- Testing of Navigation Structure
 - Simple HTML-prototype without graphics(60 pages functional prototype)
 - Whole structure(Covered all main areas)
 - Test persons of three main user groups(3x10 test persons: patent attorneys(专利代理), patent applicants(申请者), generally interested persons)
 - 25-30 search tasks(3 realistic task sets)
 - Automatic logging of search paths
 - Qualitative interviews



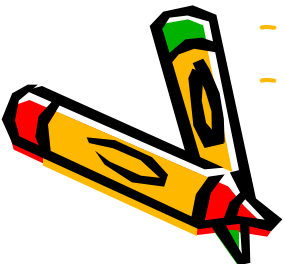
Usability testing

- Focus:
 - Identification of remaining problems
 - Empirical answers to open design questions
- Final Review
 - Final usability check of masterpages
 - Monitoring of changes
 - Check of Integration
 - Big Picture
- Methodical Note:
 - Done by usability engineers not involved in the project until then



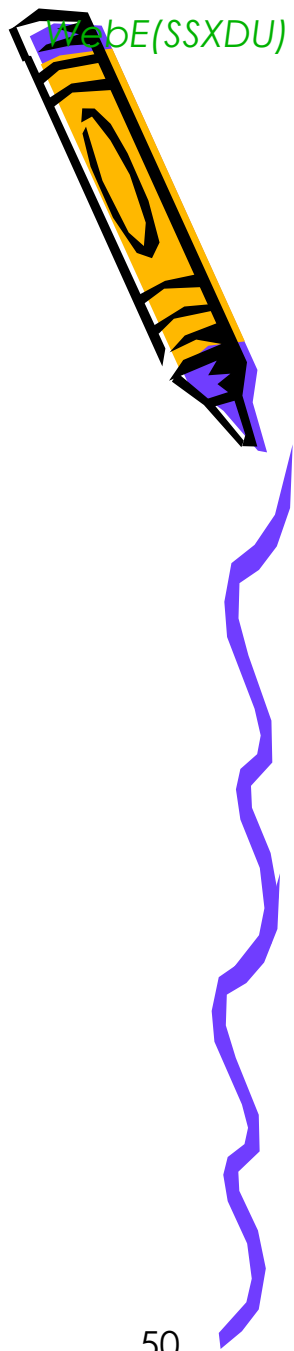
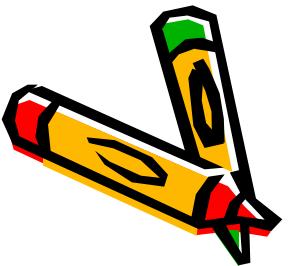
Usability testing

1. Choosing your subjects
2. Before the usability testing
 - 轻松的场所、必要的会见和通知、让参与者知道时长、任务类型、签约
 - 不要使用“可用性测试”或“市场调查”等术语
3. Beginning the usability testing
 - 让测试者熟悉环境（网站的名称、URL），获取他们的喜好信息
4. Choosing tasks
 - 如某电子商务网站：购买、付账单、联系顾客
 - 你的目标观众是不是在做你希望的事情
5. How to word tasks (写任务)
 - 更自然的行为：情节（而非任务指令）
6. Presenting(呈现) tasks
 - 一次只给参与者一个任务或者改变测试方式
7. How to behave (执行) during the usability testing
 - 过程控制：除了诱惑，保持安静和视而不见、你的所有反馈都必须中立。
8. After the usability testing
 - 汇集信息：对网站的整体印象、判断是否已达到预期的期望
 - 从参与者对网站结构和功能的记忆中去判断网站的结构的逻辑性并帮助识别任何你可能忽略的分类标签

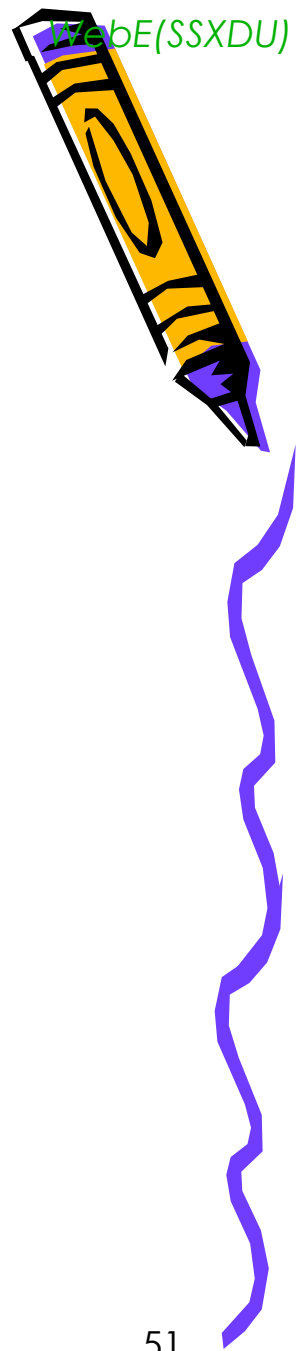


性能测试方法

- 虚拟用户方法
- WUS方法
- SPE方法



安全性测试



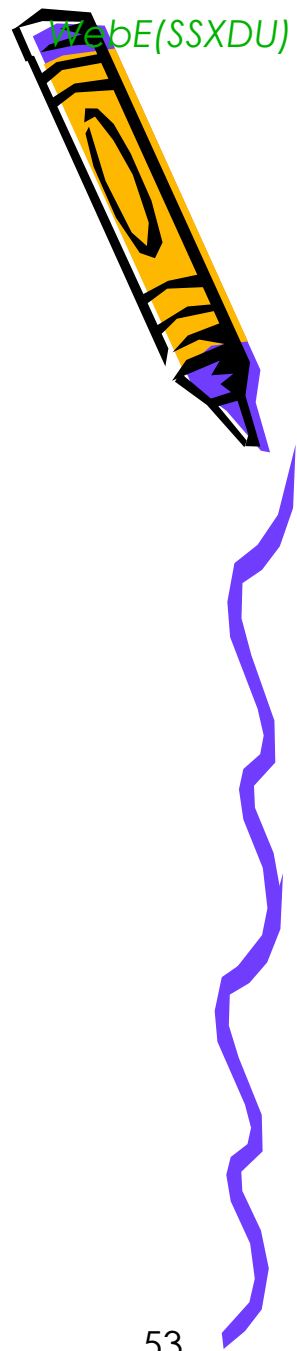
Security Testing

- Security encompasses...
 - Confidentiality(机密性)
 - Authorization(授权, 认可)
 - Authentication(认证)
 - Accountability (责任性)
 - Integrity(完整性)
- A systematic test scheme is strongly encouraged.
- Testing for correctness is not sufficient
 - Is confidential data inadvertently(因疏忽所致地) exposed?
 - What happens if we input incomplete data?
 - What happens if we inject malicious code?

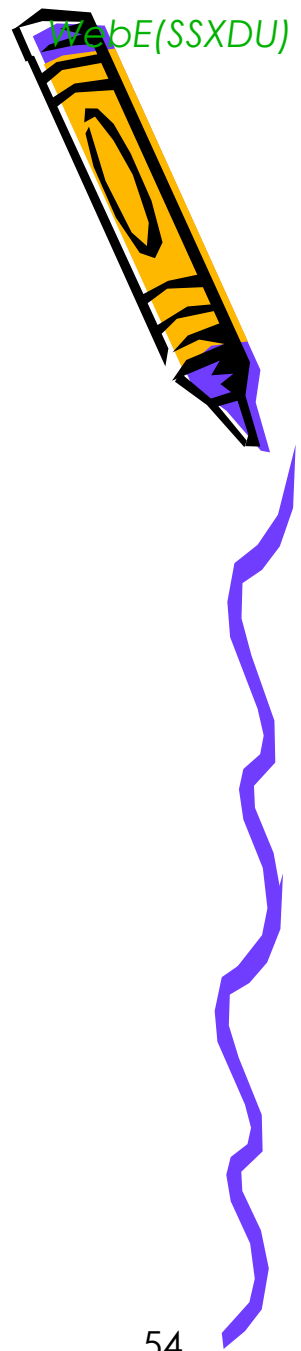


安全性测试

- 数据加密测试
- 用户身份验证测试
- 日志文件测试
- Session测试
- 备份与恢复测试
- 访问控制策略测试
- 安全漏洞测试
- TCP端口测试
- 服务器端脚本漏洞检查
- 防火墙测试

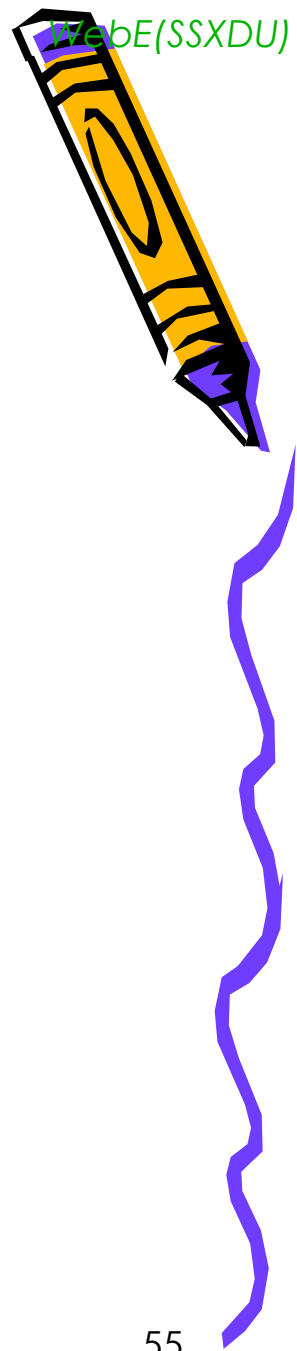


接口测试

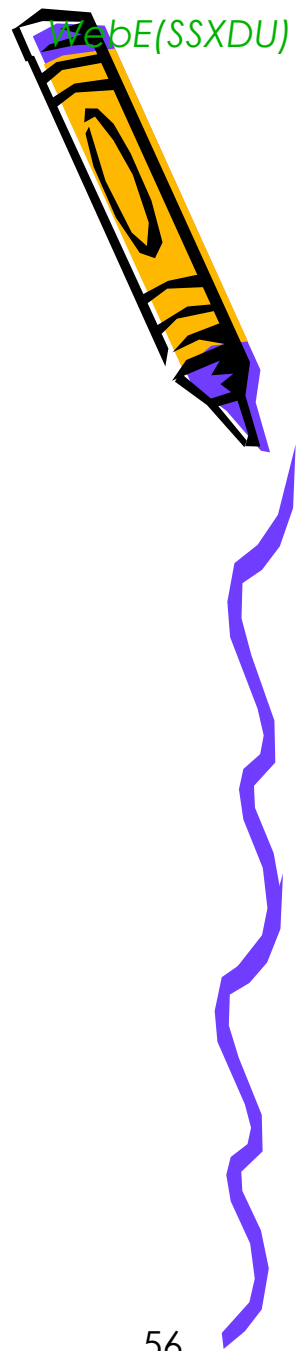


接口测试

- 服务器接口测试
- 外部接口测试
- 错误处理测试



WEB服务测试

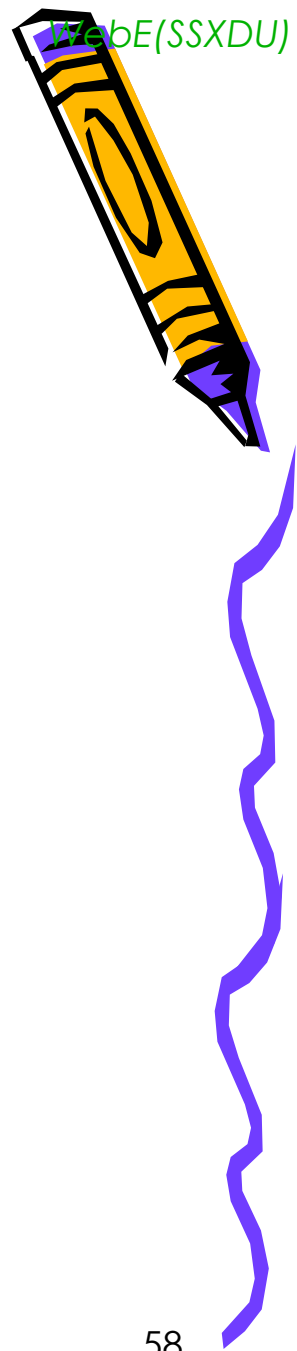


Web服务测试特性

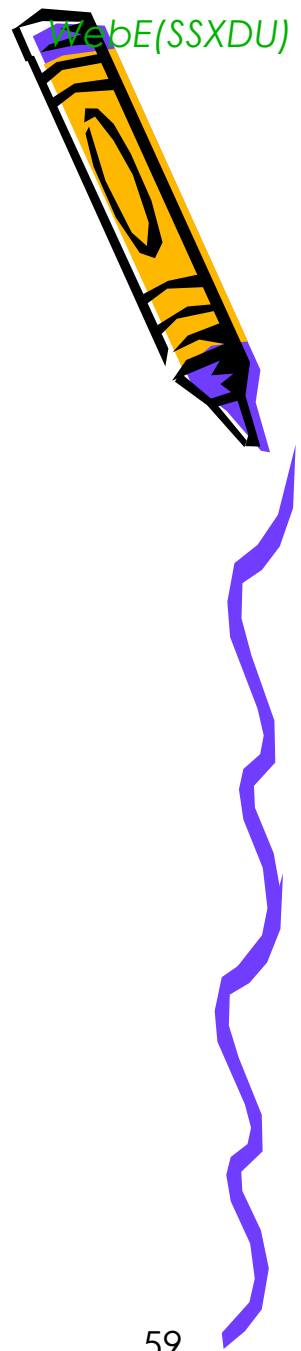
测试因素	Web服务测试
测试参与者	需要服务的提供者、代理和请求者共同参与到测试过程中
测试分布	分布、远程、多阶段
测试模型	不拥有服务代码（服务开发者除外），有限的测试模型（可控制性和可观测性低）
测试覆盖	除服务开发者拥有传统的覆盖范围外，服务提供者、服务集成者、服务请求者和服务代理可能只有黑盒覆盖范围
测试执行	在线、及时测试
测试客户端	需要构建测试客户端调用被测服务
测试预测	Web服务动态绑定，很难预测其实际运行情况，难以生成测试预测
回归测试	在线、基于运行时收集数据的测试；除服务的开发者和提供者外；服务请求者、服务集成者和服务代理并不掌握服务的演化情况，回归测试需要额外的信息支持

Web服务测试内容

- Web服务基础设施的验证与确认
- 独立的Web服务测试
- 集成的 Web服务测试

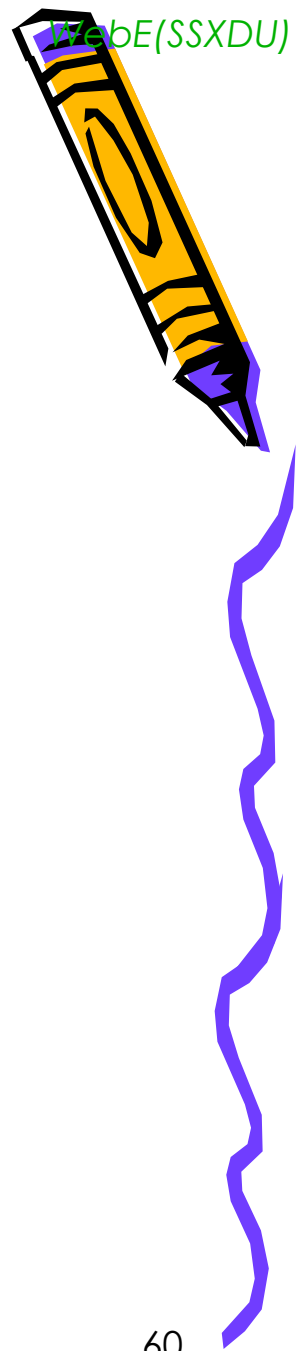


测试工具



测试工具的功能

- 测试计划与管理
- 测试用例的设计
- 静态和动态分析
- 自动运行测试
- 系统监控

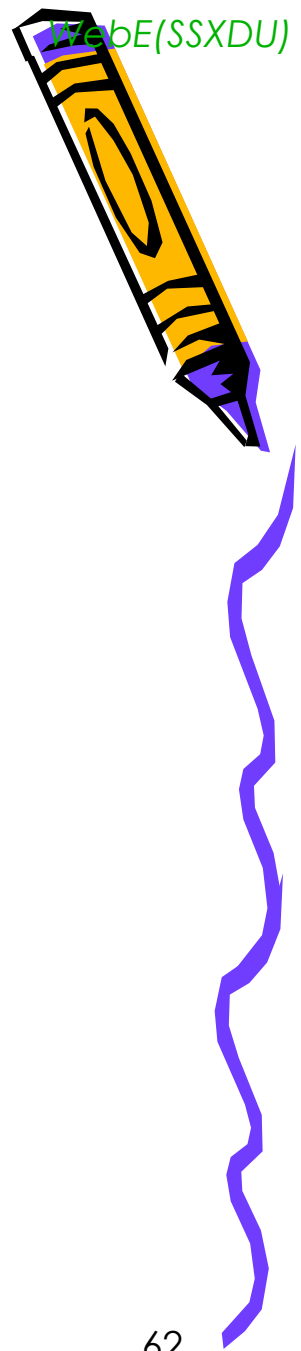


测试工具

- 功能测试工具
 - 如：WinRunner、Rational Robot等
- 性能测试工具
 - 如：JMeter、LoadRunner和WAS等
- 安全性测试工具
 - 如：IBM Rational Appscan、HP WebInspect等
- Web服务测试工具
 - 如：SOAtest, SOAP UI等
- WebUI
 - Selenium, Dagger



总结与展望



总结与展望

- Web应用测试的目标是对系统的质量维度进行检查，找出隐藏的错误或缺陷。
- 主要针对的是Web应用的内容、功能、结构、可用性、导航性、性能、兼容性、互操作性和安全性属性。
- Web应用的动态性越来越强，许多页面内容都是动态生成的，这就对测试中动态分析技术提出了更高的要求，需要提供更为有效的动态分析手段。
- 目前的自动化测试技术和工具还存在亟待解决的问题，最为典型的的就是测试数据的自动生成问题，所以自动化测试技术还将是Web测试的一个重要研究方向。



Project Task: Task8

- Web应用测试
 - 对构建的Web应用（部分功能）进行测试。

