第一图书网, tushu007.com

<<精通ASP.NET Web程序测试>>

图书基本信息

书名：<<精通ASP.NET Web程序测试>>

13位ISBN编号：9787302247845

10位ISBN编号：7302247846

出版时间：2011-3

出版时间：清华大学

作者：(美)麦克怀特//霍尔|译者:钱峰//张少华//丁峰

页数：393

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

近年来，软件质量有所提高，这都要归功于软件团队耗费了无数个小时对软件进行反复测试，以确保在软件到达用户之前能够找到并消除各类问题。
对于asp．net
web应用程序执行的测试存在多种类型，《精通asp．net
web程序测试》将告诉读者如何执行这些类型的测试，包括应用程序的功能测试、安全性测试、负载／压力测试、一致性测试和町访问性测试。
拥有了《精通asp．net
web程序测试》，就拥有了测试使用asp．net系列创建的web应用程序的最佳测试方法。

主要内容
?·唯一一本涵盖了当前非常流行的asp．net语言所有测试方面的专业书籍
·为测试应用程序时可能遇到的问题提供提示信息
·指导读者使用各种方法和工具成功地测试应用程序
·解释测试的基本概念，引导读者了解每种测试学科
·让读者接触到应用程序测试领域中最新且最常用的术语

作者简介

Jeff McWherter是位于美国密歇根Okemos的Web Ascender的负责简化工作的总监。
他毕业于密歇根州立大学，获得电子通信学学位，并在软件开发方面拥有14年的专业经验。

他是Greater Lansing User Group for.NET（GUJGnet）项目的创始成员和现任总监。
他喜欢分析代码，应用设计模式，寻找模糊名称空间，以及在公园散步。
他对编程的终身兴趣始于1983年的Home ComputingMagazine中一篇使用BASIC语言编写的名为Boa Alley的相关游戏文章。

Jeff目前居住于密歇根Lansing附近的农庄。
在工作之余，他喜欢与美丽贤惠的妻子一道攀岩和攀冰，这激发了他最喜爱的一项业余爱好——旅行。

Ben Hall是来自英国的一位充满激情的软件开发和测试人员。
Ben喜欢探索测试软件的不同方式，重点研究如何最有效地测试不同类型的Web和桌面应用程序。
他还热衷于软件的开发，使用ASRNET和Ruby on Rails开发Web应用程序。
Ben是微软C#领域的MVP。

书籍目录

第1章 初步关注点
1.1 测试工具的历史
1.2 测试术语
1.3 测试误区
1.4 迭代过程
1.5 学习测试的理由
1.6 从哪里开始
1.7 应当何时使测试自动?
1.8 asp.net系列
1.9 本章小结
第2章 设计和可测试性
2.1 开发人员应当关心测试的理由
2.2 开发人员测试
2.3 单元测试
2.4 单元测试架构
2.5 设计和测试驱动开发
2.6 如何应对设计变更
2.7 本章小结
第3章 单元测试和测试驱动开发
.3.1 asp.net webform
3.2 单元测试asp.net webform
3.3 模型-视图-控制器模式
3.4 实现比萨店的应用程序
3.5 本章小结
第4章 集成测试
4.1 集成测试与单元测试
4.2 外部系统
4.3 本章小结
第5章 自动化用户界面测试
5.1 实现自动化用户界面测试的方法
5.2 功能测试
5.3 自动化用户界面测试的重要性
5.4 用户界面自动化的问题
5.5 用户界面测试工具
5.6 使用记录和回放自动化用户界面
5.7 用户界面测试模式
5.8 作为对象模型自动化用户界面
5.9 更多watin示例
5.10 常见的watin错误
5.11 测试javascript
5.12 自动化跨浏览器测试
5.13 自动化程度
5.14 将自动?用户界面测试应用到asp.net系列中
5.15 本章小结
第6章 验收测试

6.1 验收测试术语
6.2 本章小结
第7章 手动测试
7.1 手动测试如此重要的原因
7.2 部分自动化
7.3 不只是简单地破坏应用程序
7.4 破坏应用程序
7.5 手动测试的实用工具
7.6 跨浏览器测试
7.7 测试javascript
7.8 手动测试用例
7.9 本章小结
第8章 性能测试
8.1 性能测试基础
8.2 性能测试的重要性
8.3 捕获性能需求
8.4 容量规划
8.5 负载测试模式
8.6 云计算中的性能测试
8.7 负载测试工具
8.8 本章小结
第9章 可访问性测试
9.1 可访问性误区
9.2 可访问性测试的重要性
9.3 残疾人简介
9.4 创建可访问的站点
9.5 可访问性标准
9.6 可访问性测试
9.7 可访问性测试工具
9.8 辅助技术
9.9 帮助检查可访问性的工具
9.10 具栏
9.11 将可访问性测试应用到asp.net系列中
9.12 本章小结
第10章 安全性测试
10.1 安全性条款
10.2 安全性测试的重要性
10.3 安全性测试的困难
10.4 条例和政策法规
10.5 安全指南
10.6 漏洞评估
10.7 工具
10.8 可利用的资源
10.9 本章小结

章节摘录

会话超时是大量应用程序中的一个问题。
开发人员经常忽略人们会在后台打开浏览器窗口，而跑去做其他事这一事实。
他们将信息存储在一个会话中，并希望这些信息一直在那里。
遗憾的是，事实并非如此。
这些问题对于某些应用程序尤为重要。
但是，应当为所有ASP.NET Web站点考虑这一点。
另外，可用性测试还有一个重要问题。
一般来说，测试人员和开发人员花费大量的时间和精力在产品上，因此，往往自动忽略“已知错误”或“设计”的一些问题，因为团队内部之前讨论过，他们已经对此有了认识。
基于这个原因，让第三方参与，并以新鲜的视角协助可用性测试会是一个很好的主意。
这些第三方不应当执行所有任务。
但是，他们能够协助通用流程，测试应用程序的组织方式是否是可理解的。
通常，结果会非常有用并且有趣。
7.3.2 文档　　除了可用性以外，文档也需要测试。
有两种主要类型的文档。
一个是站点可能会提供的一般帮助会话，另一个是自己在Web站点上找到的一般性措词、方向和文本。
在第一次测试站点的实际帮助会话时，最重要的是记住让这个帮助文档实际与用户界面相匹配。
很多次，在使用帮助系统时候，发现它们都只是针对该站点的一个旧版本。
应当更新所有屏幕截图或指导步骤，用以反映该应用程序的最新版本。
需要测试这一点来确保其实现。
另一个要测试的是，这些步骤能够实际上解决问题并帮助用户。
很容易错过一个步骤，或是提供了一个错误的代码示例。
只需要像用户那样遵照指南一步步操作，从而确保所需的一切都得以涵盖。
最后，需要验证技术术语是否以正确的方式用在了正确的地方，从而不会导致用户的混淆。
在某些情况下，技术性的措词可能导致更多的问题，因此需要根据目标用户来决定是否需要删除这些措词。
文档的另一部分是站点的通用措词。
站点上文本正确无误，准确，没有拼写错误是很重要的。
确保这些是正确的，并且这些文本能够在用户需要的时候帮助他们，这是测试人员应该做的。
我们发现，用户一般扫描式地阅读Web页面，这也包括文档。
因此，请试着让Web页面便于浏览。
7.3.3 错误消息　　类似于文档的方式，错误消息也需要进行测试。
关于错误消息最重要的一点是，它们需要有用。
错误消息，简单地表述为“很抱歉，发生错误了”，这将惹恼最终用户。
这种说法没有向用户提供任何有关失败的原因，是谁的过错，或者如何恢复这些方面的解析。
错误是一个事实，它们会发生，但重要的是，这个站点如何处理这些问题。
最好的错误消息是向用户提供一些有用的关于错误原因的信息。
它们还应当指导用户在错误发生之后的下一步动作。
在错误发生之后，用户是完全不知所措的，并被带出了那种舒服地使用该应用程序的境地。
错误消息应该明白这一点，并指导用户进行下一步的操作，并且如果有可能的话，解决这个错误。
通过帮助用户解决这个错误，会让他们感到很高兴，因为这让他们自己能够避开这个问题。
如果站点只是彻底失败，没有任何帮助或信息来指导如何处理，那么用户将感到很困惑，甚至可能直接关闭浏览器，站点将失去访问者。
如果能够帮助他们处理之前的错误并让他们呆在站点上，就留住了访问者。

错误消息永远不该做的事是让用户感到自己愚蠢。
即使是用户自己的错误，也不应当写在错误消息中，因为这可能会对站点起到负面作用，由于此时站点已经失败在先，用户可能已经不高兴了。
手动测试确实包含了旨在不破坏应用程序的任务，它的一项最重要的任务之一是查找实际故障。其中最强大的技术是探索性测试。
7.3.4 探索性测试　探索性测试包括认识应用程序内部的工作原理，开发测试用例以及作为一个独立过程执行测试用例。
设计探索性测试的目的是通过以各种方式使用应用程序，来让测试人员学习了解这一应用程序，旨在找到故障和错误。
虽然测试人员在探索性测试阶段使用这一应用程序，但他们却正在逐渐理解输入如何通过该系统影响到结果的核心部分。
通过探索不同的选项，尝试不同的系统，他们应该能够找到问题，或者那些看起来不对的系统部分，可能是呈现或者计算结果。
这个探索过程非常重要。
在编写自动化测试用例时，一般非常注重某个特定输入得到一个预期的输出。
……

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com