

# 西安交通大学

## 软件质量保证与软件测试课程论文

2018-2019 年春季学期

---

### web 开发中的测试技术最新进展综述

姓名: 张超峰

学号: 3118311115

日期: 2019 年 5 月 8 日



# web 开发中的测试技术最新进展综述

## 1. 摘要

在过去的十五年中，Web 应用程序已经从早期的，简单的和基于超文本的应用程序演变为新一代的更复杂的，交互的，和自适应的应用程序。基于 Web 的系统开发的新范例，新体系结构和新技术不断涌现。在这种背景下，越来越多的人提出了用于有效测试 web 应用程序的新技术和新工具。本文报告了过去几年中出现的，以 Web 应用程序测试为主题的一些相关文章，并讨论了该领域的一些未来趋势。

## 2. 引言

随着个人电脑和智能手机的普及，越来越多的人都成为了互联网用户。随着电子商务和电子支付等技术的发展，同时还有客户与互联网的互动，迫使 Web 应用程序开发人员需要不断的提高应用程序的质量，以满足客户的需求。但是，由于 Web 应用程序的多功能性，动态性和复杂性，使得 Web 应用程序的测试成为一项非常具有挑战性的任务。

一方面，由于 Web 应用程序的特殊性，一些已经应用于其他传统软件应用程序的测试技术无法和 Web 应用程序一起使用。这些功能包括：用户可能具有不同熟练程度和背景，应用程序的执行流程，不同的客户端环境，浏览器兼容性问题。

另外一方面，即使网站在浏览和功能等技术上没有问题，其他的一些错误也会很容易引起用户的挑剔。诸如死胡同，缺失链接，语法和拼写错误，导航不良，界面复杂，浏览器不兼容，图形加载时间和缺乏搜索功能等问题。是 Web 测试，成功和接受用户的重点领域。

此外，其他挑战还包括 web 应用程序的即时性、可用性、可靠性、缺乏文档、缺少测试目标和测试成本，这些情况使得 web 测试任务更加繁琐和困难。

今天，大多数交互式 Web 应用程序都是使用 Web 2.0 技术设计的，其中有许多都依赖于 JavaScript。JavaScript 能够通过动态修改 Web 页面的内容和结构来开发客户端功能，而无需调用服务器。这意味着服务器端测试框架必然无法覆盖完整的应用程序行为。

这篇文章调查了一些过去几年中出现的，以 web 应用程序测试为主题的文献，主要包括有：web 应用程序测试用例的生成、web 回归测试、web 一致性测试、web 可用性测试、验收测试、web 漏洞安全测试、浏览器兼容性测试、web 自动化测试等方面的最新进展。

## 3. web 开发中的测试技术

### 3.1. 回归测试

#### 3.1.1. 简介

回归测试是一种非常常用的软件测试方法。但由于 web 应用程序更新频繁，使得它在 web 应用程序上的使用受到了很大的限制。

由于安全攻击，功能更新或用户首选项更改，Web 应用程序会经常更改和升级。这些修复通常涉及小补丁或修复 bug。但测试人员仍需要对其产品执行回归测试，以确保更改未引入新错误。然而，将回归测试应用于整个产品的代价很昂贵。随着应用程序的增长，

测试用例数量增加，在指定的时间内执行所有测试用例变得非常困难。因为通常发布补丁的周转时间都很短，通常情况下没有这么多时间。

### 3.1.2. 方法

A. Kumar 等人在[1]中提出了一个新的，可以在 Web 应用程序上使用的回归测试方案。一种以依赖事件驱动技术为基础，创建原始和修改后的 Web 应用程序事件依赖图，然后将原始和修改后的 Web 应用程序图转换为事件测试树。先删除循环冗余，然后比较两个事件测试树，然后识别出已更改的节点和可能受影响的节点。通过已更改和可能受影响的节点来选择用于 Web 应用程序回归测试的测试用例，最终可以大大减少回归测试中测试集大小。

D. Garg 等人在[2]中建议使用基于功能依赖图 (FDG) 和程序间控制图(ICG)的两级优先级排序方法。先使用 FDG 捕获 Web 应用程序中功能模块之间的功能依赖性。然后对 FDG 中的每个功能模块使用 ICG。其中 FDG 用于对测试集进行优先级排序，而 FDG 的功能模块内的 ICG 用于对该测试集内的测试用例进行优先级排序。

他们提出了九种不同的优先级策略来优先考虑 Web 应用程序的功能测试套件。同时，他们还提供了一种自动化方法，用来自动选择与新修改功能相关的测试用例的优先级。

A. Marback 等人在[3]中提出了一种新的回归测试方法，该方法适用于经常修补的 Web 应用程序。我们的方法通过使用影响分析的代码更改，来识别受影响的区域，并通过改变正在使用的程序切片，为那些受影响的区域生成新的测试用例。

作者实现了一个 PHP 分析和回归测试引擎 (PARTE)，并使用开源 Web 应用程序执行了受控实验。结果表明，与控制技术相比，基于程序片的测试用例生成方法可以生成测试程序的修改版本所需的测试用例少得多。可以有效减少为修补 Web 应用程序软件执行回归测试所需要的时间。除此之外，它还可以节约测试人员的时间和精力，为新的测试用例和其他相关工作挤出了时间，使得测试人员在测试主要版本时提高测试效率。

可以有效地降低频繁修补的 Web 应用程序的回归测试成本。

## 3.2. 一致性测试

### 3.2.1. 简介

一致性测试是提高软件可靠性和确保软件质量的方法之一。通常，一致性测试是通过测试人员运行程序，查看被测试的应用程序是否按照符合其要求的规范来执行。

现如今，动态网页，会话/cookie 技术以及 Web 浏览器提供的功能，使得 Web 应用程序更加复杂。即使 Web 应用程序的设计都是正确的，但是，当它部署到支持环境中时，其功能依然可能会脱轨。因此，我们不仅要检查规范是否符合其实现，还要检查实现是否符合规范。

### 3.2.2. 方法

S. Chen 在[4]中提出了一种通过使用形式化方法来规范 Web 应用程序规范及其实现的一致性测试方法。为了测试应用程序的实现是否符合其规范，测试规范是否符合其实现。作者还提出了一种基于模型检查的方法来自动执行 Web 应用程序的一致性测试。通过这种方法，一致性测试可以自动执行。

L. Li 在[5]中研究了如何使用实时序列图 (LSC) 来构建 Web 应用程序的可执行模型。LSC 是一种富有表现力的基于场景的规范语言。采用上下文无关语法 (CFG) 来描述系统的外部事件，并引入模拟树来检查 LSC 的运行是否由与 CFG 解析相对应的事件触发。如果一致，模拟结果将生成状态转换系统 (STS)，否则，跟踪失败用例作为测试反例。使用基于 STS 生成的测试用例，是基于一致性模型的有效测试用例。

使用 LSC 为 Web 应用程序生成基于模型仿真的一致性检查和测试用例，为生成一组有效的测试用例提供了一种新的自动化方式。这项研究的优势在于我们的模拟器不仅可以确保设计模型在系统开发的早期阶段与规范一致，而且还可以减少生成的测试用例数量。该研究在一定程度上保证了系统的一致性，缩小了测试用例数量，生成了有效的测试套件，降低了测试成本。

### 3.3. 可用性测试

#### 3.3.1. 简介

大量用户的参与使得可用性成为 Web 应用程序成功的关键因素。提高 web 应用程序的可用性有许多优点。根据施乐公司[9999999]进行的一项调查，提高可用性在质量，产品，成本和客户满意度等方面产生了很多好处。包括以下几点：

- 通过更高效的设计方案和减少代码修订来提高生产力。
- 识别早期开发过程中的设计问题，节省时间和金钱。
- 减少培训需求和提高用户生产力。
- 提高客户满意度。

可用性大师 Jacob Nielsen 根据五个组成部分定义了可用性：可学习性，效率，可度量性，错误数量和用户满意度[6]。

可以使用不同的方法来检查应用程序的可用性。但是在软件开发领域还没有建立最有效的、最具成本效益的可用性评估方法。由于时间和成本限制的原因，广泛的可用性评估是不可行的，因此如何在有限的时间和成本下提高可用性，达到足够的水平是这一领域的关键问题。

#### 3.3.2. 方法

现在实际上被 web 行业所使用的可用性测试方法有五种：远程可用性测试，教练方法[7]，共同发现学习，性能度量和 Think Aloud Protocol 方法：

远程可用性测试：当用户和可用性专家在物理距离上相距比较远时，这种方法成为最终的选择。远程可用性测试可以跨越组织和地理边界，并支持软件开发的新方法，例如：全球外包和开源软件开发。

教练方法[7]：在教练方法中，可用性专家作为教练，鼓励用户在可用性方法中向可用性专家提问。可用性专家以适当的指示做出回应。通过听取用户的提问，来确定问题并帮助设计文档。

共同发现方法[6]：这种方法需要两个参与者，他们一起工作并在讨论产品界面时各自用语言描述他们的想法，这种方法还可以发现特定任务的完成方法。

性能测量[8]：在此方法中，需要参与测试的人获取执行任务时相关的定量数据，这些定量数据将会在比较测试或基准测试中起到很大的作用。

Think Aloud Protocol[8]：在 Think Aloud Protocol 方法中，要求用户在执行任务时说出自己的想法。通过在完成任务的过程中尝试的说出他们的想法，用户可以说明他们完成任务时的想法。这有助于澄清他们在此过程中遇到的任何困难。

## 3.4. 验收测试

### 3.4.1. 简介

验收测试是软件开发生命周期中必不可少的活动。其目的是验证软件是否满足客户要求。自动验收测试也可以作为回归测试套件的一部分，以验证软件变更不会违反客户的要求。

### 3.4.2. 方法

N. Negara 在[9]中提出了一种 JavaScript Web 应用程序的自动验收测试方法，该方法使用易读的、直观的脚本语言，我们的方法允许用户在测试脚本中描述用户需求，然后使用自动网站爬虫在 Web 应用程序上执行这些测试脚本。

[9]中所提到的方法依赖于 Crawljax[10]，这是一个开源的爬虫项目，是一种基于 Ajax 和 JavaScript 等 Web 2.0 技术的现代 Web 应用程序。使用[9]中所提出工具 CrawlScripter，可以很容易的让开发人员创建与客户用例相对应的，易于理解的自动验收测试，并通过运行应用程序来执行它们。

## 3.5. web 导航测试

### 3.5.1. 简介

Web 应用程序的导航可以定义为：用户可以访问的，以执行所需功能的网页序列。可以通过测试链接（静态和动态），表单，客户端和服务端以及业务工作流程来测试导航行为。

### 3.5.2. 方法

为了测试导航行为，从模型生成测试序列（用户访问的网页序列），P. Bansal 在[11]中提出了一种基于模型的导航测试用例生成技术，该模型可以用来描述 Web 应用程序的导航行为，其中表示 Web 应用程序的导航行为的模型是从需求中导出的。遍历该模型可以生成测试序列，遍历的方法可以从主页开始访问，并返回主页的所有可能的网页序列。除此之外，由于存在可能导致路径爆炸的环路，在[11]中作者提出了一个解决路径爆炸问题的循环测试标准，并展示了如何使用所提出的标准来生成覆盖所有独立路径的测试序列，至少覆盖每个节点和每个边缘一次。该测试序列随后可以与输入数据合并用来生成测试用例。

## 3.6. 漏洞安全测试

### 3.6.1. 简介

当今最紧迫的安全问题之一是 Web 应用程序漏洞，例如：跨站点脚本（XSS），SQL 注入等。

Web 应用程序一直都在面临被攻击的威胁，这与它们所处的运行环境具有很大的关系。Web 的日益普及使这些应用程序变得越来越复杂，但它们所提供的服务和数据，不管对与用户，还是潜在的攻击者，都是宝贵的资源。因此，web 应用程序的安全性至关重要。敏感的数据一旦被泄露，往往会产生非常严重的后果。

Web 应用程序最突出的一类漏洞是跨站点脚本（XSS）。XSS 漏洞是由于缺少或没有充分验证的输入数据，因此攻击者可以提供将恶意脚本或 HTML 片段注入到 web 应用程序中，受害者在浏览器上浏览这些网页的时候，这些恶意脚本或 HTML 片段就会被执行。就可能会造成数据泄露或者用户会话被劫持。

### 3.6.2. 方法

A.Vernotte 在[12]中提出了一项关于通过扩展 MBT 流程为 Web 应用程序生成漏洞测试用例的研究。其中生成过程不是由模型驱动，而是由测试目的驱动。为了生成测试用例，生成引擎基于表示系统行为和外部恶意刺激的建模。这样生成引擎可以专注于 Web 应用程序的特定部分或特定类型的漏洞，和其他的漏洞发现技术相比，更具有优势。

A. Avancini 在[13]中提出了一个针对 Web 应用程序的安全性测试方法，通过结合使用静态分析，遗传算法和约束求解来生成测试用例。并且将遗传算法与基于求解器的局部搜索策略相结合。在[12]中，当遗传算法陷入局部最优并且无法进一步改进时，局部搜索求解器就会开始运行。

## 3.7. 浏览器兼容性测试

### 3.7.1. 简介

由于 IE 的主导地位被打破，浏览器市场竞争激烈。用户在浏览器的选择上具有很大的灵活性，这导致了一个新的问题，称为跨浏览器兼容性（CBC）问题。

CBC 问题与网络浏览器的发展几乎同时发生。同一网页在不同浏览器（和跨平台）中出现不一致的结果。不兼容的问题可能会导致页面出现错误的结构，甚至是严重的功能故障。保持目标浏览器的一致性应用程序开发人员的一项重要任务。

Web 应用程序开发人员需要在完成敏捷应用程序开发任务的同时应对浏览器兼容性的压力，并将其快速推向市场。开发团队不得不花费巨大的人力成本来进行 CBC 测试并破解错误，这使得 Web 应用程序的开发变得更加困难。

### 3.7.2. 方法

X. Li 在[14]中提出使用一种交互转换模型（ITM）来表示包含交互的网页的行为。ITM 将交互元素视为纯独立实体，并将其相关系数降低为零，但在现实世界中，它们之间也存在约束或时序关系。考虑到互动元素之间的时间关系将使 ITM 更加完善，但也会使模型更加复杂，有可能会状态爆炸的问题。

## 3.8. 自动化测试

### 3.8.1. 简介

如今，大量的软件系统是基于 web 的应用程序。与任何软件一样，这些应用程序在部署到最终用户之前必须确保其质量。这种情况下，测试是保证系统质量的重要活动。作为尝试用更少的资源做更多的事情的一部分，团队希望能更充分的测试软件。为了实现这一目标，越来越多的团队正在转向自动化测试。

数据库是大多数基于 Web 的应用程序的核心。这些应用程序与数据库之间的交互在软件项目的测试中具有一些困难。不同软件版本之间数据库中状态/内容的差异可能有助于检测出回归测试中潜在的失败。但是，手动验证这些差异是一项艰巨而耗时的活动。因此，自动化地检查数据库中的数据可以提高执行此任务的速度。

### 3.8.2. 方法

A. de Castro 在[15]中介绍了一个 Selenium RC 工具的扩展，可以很方便的在 web 应用程序中测试数据库中的数据。其中的函数可以自动的打开和关闭数据库连接，并将测试数据与 Web 应用程序使用的数据库中存储的数据进行比较。该解决方案旨在通过减少测试过程中的工作量来提高系统质量，因为测试 UI 的工作和测试数据库中的工作是同时进行的。

自动数据库测试可用于帮助软件工程师评估基于数据库的 Web 应用程序的质量，与手动或半自动化策略相比，可以减少测试任务的工作量。

## 4. web 测试技术之后的发展方向

Web 应用程序的技术和体系结构随着时间的推移而不断发展，并且仍在不断变化。从专门为第一代静态 Web 应用程序提供的，为静态 html 资源的访问而设计的，到现在已演变为更复杂和动态的多层系统的 Web 应用程序。业务逻辑在客户端和服务端实现，并且它们之间通过异步层之间进行交互（例如在 RIA 和 AJAX 应用程序中）。此外，基于内容的管理系统，混搭技术以及现有组件和框架的新开发方法正在出现。在未来，我们可以预见，Web 应用程序将变得越来越复杂，它将集成现有服务，组件，应用程序和多媒体，并且必须能够适应不断变化的执行环境和操作环境。此外，它们必须能够被越来越多的，配备有异构硬件，不同的操作系统和执行平台的移动设备访问，这将带来新的兼容性问题。

在这些未来的场景中，需要解决新的测试问题。总的来说，由于这些应用程序的复杂性更高，手动测试将变得越来越不可行。越来越需要 Web 测试自动化的解决方案。研究人员必须认真研究特定测试技术的适用性。例如基于搜索，基于模型和基于爬虫网络的测试技术，以支持 Web 测试自动化。为了应对动态和自适应应用程序的测试问题，测试框架和环境提供稳定的运行监控功能是必要的。最终，新的测试基础架构也需要利用云计算的计算能力，需要设计为能够应对在异构执行平台上运行的测试应用程序的碎片问题。在这些情景中，需要评估过去测试技术的适用性，并且需要研究新的测试解决方案。

## 5. 总结

自从大约二十年前第一代 Web 应用程序开始普及以来，研究人员从不同的角度探讨了测试问题，并为这一领域的成熟做出了贡献。本文报告了过去几年中出现的，以 Web 应用程序测试为主题的一些相关文章，并讨论了该领域的一些未来趋势。

## 6. 参考文献

- [1] A. Kumar and R. Goel, “Event driven test case selection for regression testing web applications,” 2012, pp. 121–127.
- [2] D. Garg, A. Datta, and T. French, “A Two-Level Prioritization Approach for Regression Testing of Web Applications,” in *2012 19th Asia-Pacific Software Engineering Conference*, 2012, pp. 150–153.
- [3] A. Marback, H. Do, and N. Ehresmann, “An Effective Regression Testing Approach for PHP Web Applications,” in *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, 2012, pp. 221–230.
- [4] S. Chen, “Model Checking Based Conformance Testing for Web Applications,” in *2012 Third World Congress on Software Engineering*, 2012, pp. 51–56.
- [5] L. Li, X. Li, and S. Tang, “Research on web application consistency testing based on model simulation,” in *2014 9th International Conference on Computer Science & Education*, 2014, pp. 1121–1127.

- [6] E. Folmer and J. Bosch, “Architecting for usability: a survey,” *J. Syst. Softw.*, vol. 70, no. 1–2, pp. 61–78, Feb. 2004.
- [7] G. Riva and C. Galimberti, *Towards CyberPsychology: Mind, Cognition and Society in the Internet Age*. IOS Press, 2001.
- [8] L. Gamberini and E. Valentini, “7 Web Usability Today: Theories, Approach and Methods,” p. 311, 2001.
- [9] N. Negara and E. Stroulia, “Automated Acceptance Testing of JavaScript Web Applications,” in *2012 19th Working Conference on Reverse Engineering*, 2012, pp. 318–322.
- [10] A. Mesbah, A. van Deursen, and S. Lenseslink, “Crawling Ajax-Based Web Applications through Dynamic Analysis of User Interface State Changes,” *ACM Trans. Web*, vol. 6, no. 1, pp. 1–30, Mar. 2012.
- [11] P. Bansal and S. Sabharwal, “A model based approach to test case generation for testing the navigation behavior of dynamic web applications,” in *2013 Sixth International Conference on Contemporary Computing (IC3)*, 2013, pp. 213–218.
- [12] A. Vernotte, “Research Questions for Model-Based Vulnerability Testing of Web Applications,” in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013, pp. 505–506.
- [13] A. Avancini, “Security testing of web applications: A research plan,” in *2012 34th International Conference on Software Engineering (ICSE)*, 2012, pp. 1491–1494.
- [14] X. Li and H. Zeng, “Modeling web application for cross-browser compatibility testing,” in *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2014, pp. 1–5.
- [15] A. M. F. V. de Castro, G. A. Macedo, E. F. Collins, and A. C. Dias-Neto, “Extension of Selenium RC tool to perform automated testing with databases in web applications,” in *2013 8th International Workshop on Automation of Software Test (AST)*, 2013, pp. 125–131.