

JIYU MOXING DE Web YINGYONG CESHI

LILUN YU FANGFA

基于模型的Web应用测试

理论与方法

钱忠胜 著



江西高校出版社

JIANGXI UNIVERSITIES AND COLLEGES PRESS

基于模型的 Web 应用 测试理论与方法

钱忠胜 著

江西高校出版社

图书在版编目(CIP)数据

基于模型的 Web 应用测试理论与方法/钱忠胜著. —
南昌:江西高校出版社, 2013. 8

ISBN 978 - 7 - 5493 - 1039 - 5

I. ①基... II. ①钱... III. ①互连网络 - 应用软件 - 测试 IV. ①TP393.409

中国版本图书馆 CIP 数据核字(2013) 第 177268 号

出版发行	江西高校出版社
社 址	江西省南昌市洪都北大道 96 号
邮政编码	330046
总编室电话	(0791) 88504319
销售电话	(0791) 88513417
网 址	www.juacp.com
印 刷	天津市天办行通数码印刷有限公司
照 排	江西太元科技有限公司照排部
经 销	各地新华书店
开 本	890mm × 1240mm 1/32
印 张	5.875
字 数	180 千字
版 次	2013 年 8 月第 1 版第 1 次印刷
书 号	ISBN 978 - 7 - 5493 - 1039 - 5
定 价	30.00 元

赣版权登字 - 07 - 2013 - 418

版权所有 侵权必究

前 言

随着 Internet 的普及以及组件、中间件和 Web Services 等技术的迅速发展以及 Web 的广泛应用,Web 应用的可靠性和质量保证成为一个非常关键的问题和研究热点。而软件测试是提高软件可靠性和保证软件质量的一种主要手段。

Web 应用是一种非常复杂的、分布式的、多层结构的交互式应用,为用户提供了一种全新的部署软件应用的方式。Web 应用的异构性、动态性、连接的多样性、控制流程的可变性以及需要快速开发与发布等特性,给 Web 应用的测试带来了新的挑战。目前还没有比较系统的方法和工具对 Web 应用进行有效的测试。Web 应用的特性,要求必须对传统的测试方法进行改进或提出新的适合 Web 应用特性的测试方法。目前的测试主要依赖测试工程师的直觉和经验,Web 应用的测试被认为是一个耗时的、代价昂贵的过程。因此,迫切需要一套新的 Web 应用测试方法并实现测试的自动化。

基于模型的、模型驱动的、以测试为中心的软件开发方法和技术的兴起和应用,以及形式化验证技术的逐步成熟,使基于模型的软件测试方法与技术在近几年得到了关注。基于模型的测试是一种突破性的黑盒测试技术,它以明确描述系统预期行为的抽象模型为依据,自动

生成可执行的测试用例,产生测试脚本,执行测试并自动评价测试结果,实现测试过程的自动化。这种方法可以使得软件测试和软件设计以及实现同时进行,改变了软件工程中“现在编程、以后测试”的工作方式,使得在整个软件生命期中可以并行地进行测试工作。因此,基于模型的测试技术是实现 Web 应用测试的有效途径。但是,这方面的研究还处于方法探索阶段。

统一建模语言(Unified Modeling Language, 简称 UML) 已经成为事实上的工业建模标准,能够很好地为用户的需求建模;而有限状态机(Finite State Machine, 简称 FSM) 为测试软件的复杂行为提供了一个有效的机制,不必考虑软件的底层实现细节。

本书以 Web 应用为研究对象,从 UML 模型和 FSM 模型出发,研究一系列基于模型的测试用例生成技术,获得基于模型的 Web 应用软件测试方法,构建一个从模型出发产生测试用例的原型工具,力图实现 Web 应用测试过程的自动化。

本书共分 7 章。第 1 章是研究背景以及软件测试概述,包括软件测试基础、基于模型的测试方法、Web 应用的测试;第 2 章在分析已有的基于 UML 模型测试准则的基础上,根据场景的概念,提出了基于 UML 用例图的七个测试准则: 场景覆盖、主要场景覆盖、行为者主要场景覆盖、主要场景序列覆盖、用例覆盖、行为者用例覆盖和用例序列覆盖,并给出了这些测试准则的实际应用,另外还提出了基于 UML 对象图的两个测试准则: 方法链覆盖和 MM - path 序列覆盖;第 3 章给出了一种基于 UML 用例图和顺序图对 Web 应用进行测试的方法,根据所提出的基于 UML 用例图的覆盖测试准则,生成所需的测试用例;第 4 章详细分析了基于规格说明的若干逻辑覆盖测试准则,提出了掩盖性逻辑覆盖测试准则,并对其进行了详细分析,从谓词表达式的结构入手,分析了文字之间的约束关系、复杂谓词表达式的分解与合成、谓词表达式之间的关系,提出了全真谓词表达式覆盖、全假谓词表达式覆盖、完全子谓词表达式覆盖、唯一文字真覆盖以及唯一文字假覆盖等测试准则,给出了谓词表达式可能的错误类型,提出了最小的能检测逻辑联结词错误的测试用例集生成算法;第 5 章给出了从 FSM 模型出发自动生成测试用例的方法,重点分析了

与 FSM 有关的最常用、具有代表性的几种覆盖测试准则,包括状态覆盖、迁移覆盖、迁移对覆盖、谓词表达式覆盖、文字覆盖、受限活动文字覆盖、掩盖文字覆盖等测试准则,在此基础上提出了全循环 K 次覆盖测试准则,并提出了分别根据上述八种覆盖测试准则从 FSM 模型生成测试用例的算法;第 6 章利用组件交互自动机复合的思想,提出了动作抽象化的复合规则,给出了一种基于 FSM 模型的组合测试方法;第 7 章设计并部分实现了一个测试用例生成方法的系统原型 FSMTGEN,力求实现测试的自动化并验证所提出的测试用例生成方法。

本书的出版得到了许多人的关注与支持。谨以此书献给所有关怀和帮助过我的人。

本书在编写过程中,参阅了许多专家的学术论文,由衷感谢所有引文作者。没有他们丰富的思想和卓越的研究成果,此书不可能顺利完成。

感谢我的工作单位江西财经大学信息管理学院的领导和同事对我的支持和帮助。和谐融洽的同事关系与同甘共苦的相互扶持给予了我温馨的工作环境。

感谢我的家人对出版此书给予的理解和支持。他们的鼓励与宽容让我有勇气面对人生的不如意与困难。感谢他们出现在我的生命中,并一直陪伴着我,让我感到无比的幸福与快乐。

本书是作者所在课题组全体师生共同努力完成的成果,课题的研究得到了国家自然科学基金项目(61262010、61163007)、江西省自然科学基金项目(20132BAB201036、20114BAB211019、2010GQS0048)、江西省博士后科研择优资助项目和江西省教育厅科技项目(GJJ12743)的资助,对国家自然科学基金委、江西省科技厅、江西省博管办以及江西省教育厅的资助表示衷心的感谢。

尽管十分努力,以求本书尽善尽美,但由于作者学识所限,加之新技术层出不穷,遗漏与疏忽之处在所难免,恳请专家、同仁和广大读者批评指教。

作者

2013 年 5 月

目 录

第 1 章 绪论	/1
1.1 研究背景	/1
1.2 软件测试基础	/2
1.3 基于模型的测试方法	/6
1.4 Web 应用的测试	/8
1.4.1 Web 应用的分类与特性	/8
1.4.2 Web 应用的测试与传统软件测试的区别	/11
1.5 本书的研究	/13
第 2 章 基于 UML 模型的覆盖测试准则	/16
2.1 引言	/16
2.2 基于 UML 用例图的覆盖测试准则	/18
2.3 基于 UML 状态图的覆盖测试准则	/20
2.4 基于其他 UML 图的覆盖测试准则	/24
2.5 小结	/27
第 3 章 从 UML 用例图和顺序图产生测试用例	/28
3.1 UML 模型视图	/28
3.2 产生用例序列	/29
3.3 识别用例场景	/37
3.4 用例 StudyPlan 对应的 FSM 的文本表示	/45

3.5 相关工作 /49

3.6 小结 /51

第 4 章 基于规格说明的逻辑覆盖测试准则 /52

4.1 引言 /52

4.2 基本逻辑覆盖测试准则 /54

4.3 决定性逻辑覆盖测试准则 /56

4.4 掩盖性逻辑覆盖测试准则 /61

4.5 谓词表达式的结构分析 /67

4.6 逻辑联结词错误的检测算法 /74

4.7 进一步讨论 /78

4.8 小结 /79

第 5 章 从 FSM 模型产生测试用例 /81

5.1 相关概念 /81

5.2 FSM 的代数表示 /82

5.3 正则表达式的构造 /86

5.4 完全路径条数分析 /90

5.5 基于 FSM 模型的测试用例生成算法 /92

5.5.1 满足状态覆盖的测试用例生成 /95

5.5.2 满足迁移覆盖的测试用例生成 /98

5.5.3 满足全循环 K 次覆盖的测试用例生成 /100

5.5.4 满足迁移对覆盖的测试用例生成 /105

5.5.5 满足谓词表达式覆盖的测试用例生成 /110

5.5.6 满足文字覆盖的测试用例生成 /113

5.5.7 满足受限活动文字覆盖的测试用例生成

/116

5.5.8 满足掩盖文字覆盖的测试用例生成 /119

5.6 进一步讨论 /123

5.7 小结 /125

第 6 章	FSM 的组合测试方法	/126
6.1	组件交互自动机	/126
6.2	自动机的复合	/129
6.3	组件交互测试序列及测试用例	/131
6.4	组件交互测试的覆盖度	/133
6.5	小结	/135
第 7 章	测试用例生成工具的设计与实现	/137
7.1	测试工具的相关说明	/137
7.1.1	测试工具的总体框架	/138
7.1.2	主要数据结构	/138
7.2	测试用例的生成	/141
7.2.1	解析 FSM 模型的 SCXML 文件	/142
7.2.2	一个运行实例	/151
7.3	小结	/154
附录	FSM 的文本表示 SCXML 的规范说明	/156
参考文献		/166

第1章 绪 论

1.1 研究背景

随着网络技术和 Internet 的迅速发展,WWW 已成为人们交付各种应用的主要平台。而目前应用设计的现状是:一方面,随着应用程序由以文档为中心向以 Web 应用为中心发展,以往简单的基于 Web 页面的开发方法已经不再适用,人们迫切需要采用系统化的方法开发 Web 应用;另一方面,由于 Web 应用和传统应用相比有着本质区别,使得传统的软件工程方法在 Web 应用的开发中显得力不从心。为解决这个问题,Web 应用的工程化方法已经成为 Web 应用开发研究的热点。这些研究旨在提供全面支持 Web 应用开发生命周期的模型和方法。Web 应用的工程方法的一个主要目的是帮助设计者用结构化方法来处理 Web 应用的特性,包括链接、导航、超媒体表示等。目前 Web 应用的复杂性正日益增加,任何稍微复杂一点的系统都需要经过仔细的设计和建模,因此应该用基于模型的方法来开发 Web 应用。

为了适应 Web 应用的快速开发、部署和维护,保证 Web 应用的质量,必须实现 Web 应用测试过程的自动化。现有的 Web 应用测试支持工具,包括 link 检查工具、HTML 验证器、捕获/回放工具、安全性测试工具以及负载和压力测试工具等大都是基于程序或源代码的静态验证和度量工具,不支持对 Web 应用的功能测试^[1]。基于模型的测试为 Web 应用功能测试的自动化提供一种非常有效的解决方案。在基于模型的测试中,测试模型和基于测试模型生成的测试用例都是抽象的、独立于平台的,从而是可重用的。测试执行时通过对测试执行环境的动态配置自动产生实例化的可执行的测试包。这一特性将大大降低 Web 应用的异构性和动态性等特性所带来的测试

复杂度。基于模型的 Web 应用测试的一个关键任务是测试用例的自动生成。

目前,尽管存在一些测试工具,但 Web 应用的测试用例设计很大程度上还是根据需求或规格说明文档手工进行的,并且使用专门的执行框架(如,被测应用的测试驱动程序)执行测试。这样的过程烦琐、低效、易出错,不能保证 Web 应用的质量。基于模型的 Web 应用测试力图提供测试自动化的方法和工具集,以较低的代价提高 Web 应用的质量。在基于模型的测试用例自动生成的方法中,现有的测试准则通常是针对过程(函数或子程序)和类的,直接将这些准则应用到 Web 应用的模型上不一定合适。因此,本书研究如何对现有的测试准则进行改造并开发更加有效的测试准则,根据 Web 应用模型构造有效测试用例的技术,获得测试用例的自动生成方法。

1.2 软件测试基础

在工业生产和制造业中,测试被当做一个常规的生产活动,常常和产品的质量检验密切相关。在这些行业中,测试的含义似乎是明确的,但在计算机软件领域内则不然。比如,什么是软件测试?测试和调试是一回事吗?它们之间有什么不同?对于这些概念的不同理解可能会涉及到测试的目的和方法。

1982年6月在美国北卡罗来纳大学召开了首次软件测试的正式技术会议,关心软件测试和软件质量的研究与开发人员第一次聚会,专门讨论他们感兴趣的问题。这次会议成为软件测试技术发展中的一个重要里程碑。

目前仍然有许多人对于“什么是软件测试”持有不正确的观点,这也恰恰是不能很好地做好软件测试工作的原因。回答这一问题的典型说法有:(1)确定用户接受的可能性;(2)对照规格说明检查程序;(3)确认系统已经能够提供使用了;(4)使自己确信开发任务已经完成;(5)检验该软件的能力;(6)表明程序中错误并未出现;(7)弄清该软件不能做什么;(8)取得该软件已能工作的信心;(9)表明程序执行得正确无误等。

当然,这些说法有其正确的方面,但毕竟含有缺陷。1983年,IEEE在软件工程标准术语中定义软件测试是^[2]：“使用人工或自动手段来运行或测定某个系统的过程,其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”。这就非常明确地指出了软件测试是以“检验软件是否满足需求”为目的。G. J. Myers^[3]则认为：“软件测试是为了发现错误而执行程序的过程”。这一定义明确指出了软件测试是以“寻找错误”为目的。Myers的软件测试的定义似乎表明,测试只有在编码完成以后才能开始。

我们认为,软件测试是“为了发现并修正软件中的错误以满足用户需求而发生检验、确认和评价软件等的一系列活动”。该定义表明软件测试是一系列活动,不仅要发现错误、修正错误,还要使软件最终满足用户的需求。为了实现软件测试的目的,传统的软件测试过程遵循图1.1所示的V字型的测试流程。该流程将软件测试过程中的阶段和软件开发过程的阶段对应起来,规定只有当前一个测试阶段结束后才可以进入下一个测试阶段。

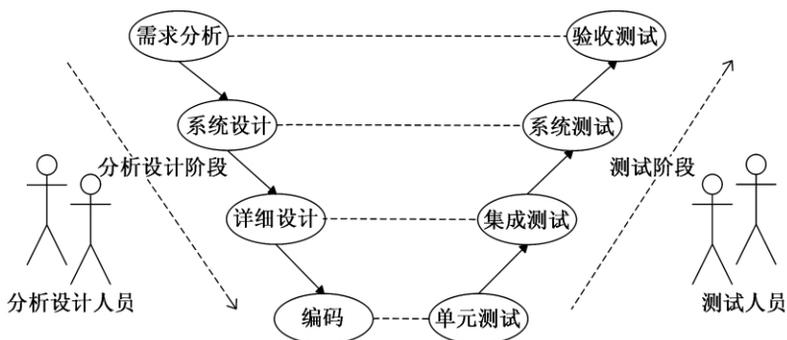


图 1.1 传统软件开发与测试的 V 模型

从这个模型中我们可以看出“单元测试”是最底层的测试活动,主要测试程序中最小的独立单元,如结构化程序中的过程或函数,面向对象程序中的类。“集成测试”的主要任务是测试单元之间的接口以保证程序单元能够正确交互。“系统测试”的主要任务是测试整个系统能否正确运行,以保证系统符合其设计模型。“验收测试”阶段主要是验证系统的功能是否符合用户需要,即是否满足软件需求规

格说明。由于不同层次的测试任务和测试目的是不同的,所以在某个测试层次很容易发现的错误可能在另一个层次就很难发现。

许多测试技术在实际运用时的一个主要障碍是,它们不同时具有有效性、应用范围的普遍性、性能鲁棒性和可扩展性等性质。有效性是指,产生的测试集的错误发现能力;应用范围的普遍性是指,在不同的应用领域,测试方法提供相同的错误检测性能的能力;性能鲁棒性是指,基于某个预先确定的有效性级别,测试方法在多种条件下重复地、一致地实施的能力(例如,若要求 100% 的分支覆盖,测试能保证一定类型或种类的错误被发现吗?);可扩展性是指,测试方法需要的测试用例个数的增长率随着软件规模(或复杂性)的增长而增加。

软件测试在当前软件开发中的重要地位仍然是其他质量保证手段所不能代替的。然而,软件测试有着一个不可克服的缺陷,即通常不可能找出程序中所有的错误,也就无法证实程序是完全正确的。

鉴于软件测试具有的局限性,一些人力图避开测试,另辟蹊径,采用程序正确性验证的方法。可是,程序正确性验证方法也是有局限的。在工业界,真正能够通过程序验证方法而得到程序正确性确认的例子是不多的。并且在大多数情况下,程序验证还仅仅停留在一些小的例子上。大中型的软件开发项目的验证无论从方法上还是验证效率上看,都还很难应用于实际。

因此,在实际的软件开发项目中,测试仍然是一种比较现实、有效的质量保障手段。如何正确地认识和对待软件测试常常是做好软件测试工作的前提。人们力图在测试中找到尽可能多的错误,尽量降低软件错误带来的危害。

在软件测试中,一旦确定好测试目标并选取好合适的测试策略之后,就需要进行测试用例的设计与构造。测试用例是按一定顺序执行的与测试目标相关的一系列测试。测试是用来诊查一个系统实现对应于它的需求规格说明所存在的错误。执行实现的所有可能场景的穷尽测试能够证明实现的正确性。可惜的是,穷尽测试是不切实际的,因为它要执行大量的或无限的场景。测试一方面能够增强实现正确性的信心,一方面由于时间和人力的约束,需要在这两个方

面有一个合理的折衷。但不管如何折衷,都需要对测试的有效性进行度量。

测试用例的设计可由软件规格说明驱动。这就意味着,高层的测试用例应该用来发现软件规格说明中反映的软件需求带来的错误,低层的测试用例应该用来发现体系结构规格说明和详细设计规格说明中所蕴含的设计决策带来的错误。一个完整的测试用例一般由两个成分组成:

(1) 测试输入。包括待测试对象读取的任何外部数据。

(2) 测试用例的期望输出。期望输出用于和实际输出做比较。

当错误被发现时,一个测试运行(test run)是成功的。这样,就获得了关于问题的额外信息和 Web 应用的状态。不成功的测试是没有发现错误的测试,被认为是“浪费时间”^[4]。这对于 Web 应用的开发更是如此,因为资源的限制和开发时间的紧迫,测试必须最小化。所以,需要尽可能早点发现严重的错误,以避免不必要的投资,因为发现错误并去除错误的费用会随着每个开发阶段的进行而急剧增加。在早期开发阶段产生的错误很难在后期阶段找到,去除这样的错误会引起广泛的变动,并需处理该错误引起的其他后继错误。

在软件测试时,测试用例的选择需要根据一定标准。1975年,J. B. Goodenough 和 S. L. Gerhart^[5]提出软件测试充分性准则的概念,指出测试准则是软件测试的中心问题。测试准则是指导测试人员产生高效的、增强软件正确性和可靠性的测试用例,是决定被测软件的测试集是否充分的准则。如果测试集不充分,必须向测试集中增加更多的测试用例,直到用例集充分时为止。测试准则不仅可以定量地规定软件测试需求,指导测试用例的选择,而且可以度量测试集揭示软件特定特征的能力,对测试结果和软件可靠性评估具有重要作用。一个好的测试充分性准则应具有如下性质^[6-7]:

- 复杂性: 软件越复杂,需要的测试用例就越多;
- 可作用性: 对任意程序均存在充分的测试数据集;
- 单调性: 如果测试数据集对一个软件系统来说是充分的,那么再增加一些测试数据也是充分的;
- 重命名性: 充分测试的软件被系统地重命名后,也是充分测

试的;

- 非外延性: 语义相同的两个程序必须进行不同的测试;
- 非复合性: 即使一个测试数据集对软件的所有组成是充分的,也不能保证对整个软件的测试是充分的;
- 空集不充分性: 空测试对任意程序都是不充分的;
- 不可分解性: 如果一个软件系统整体得到充分测试,并不保证系统各个部分都已经得到充分测试。
- 非穷尽可作用性: 一些程序存在有限的测试数据集;
- 一般多重修改性: 语法结构相同的程序也必须进行不同的测试;

从根本上来说,软件测试的核心问题是如何从被测软件无限的输入状态空间中选择有效的元素(测试用例)以满足测试需求。测试准则可以被看成是根据测试需求来划分输入空间的一种方式。

1.3 基于模型的测试方法

根据测试用例的来源,软件测试可以分为基于程序代码的测试和基于规格说明的测试两类。基于程序代码的测试从源程序出发产生测试用例,基于规格说明的测试从软件的规格说明推导测试用例。基于规格说明的测试直接根据程序的需求规格说明来确定测试用例,用测试用例运行程序产生实际的输出,再将实际输出与期望输出进行比较,以此推断程序是否有错误。这两种测试方法各有长短,互为补充。

基于程序代码的测试通常采用白盒测试方法,主要是用于程序的单元测试,可以提供具有很高的代码覆盖度的测试用例。但基于程序代码的测试的一个最大缺点是无法确定“程序是否实现了软件需求”,并且跟具体的编程语言有关。

基于模型的测试可以用在单元测试、集成测试和系统测试中,可以用来确认“程序正确地实现了其需求”,因为基于模型的测试可以发现程序没有实现的功能,并且跟具体的编程语言无关。同时基于模型的测试还为测试用例的重用提供了可能。

软件模型是对软件行为和软件结构的抽象描述。基于模型的软件测试的特点是:在产生测试用例和进行测试结果评价时,都是根据被测应用的模型及其派生模型(一般称作测试模型)进行的。基于模型的测试最初应用于硬件测试,广泛应用于电信交换系统测试,目前在软件测试中得到了一定应用,并在学术界和工业界得到了越来越多的重视。基于模型的软件测试要求充分理解被测软件。根据软件的需求构造可以用于测试的模型,其工作主要是根据测试目的确定测试对象和测试特征,针对被测软件的相关属性建立相应模型。

现有的软件规模越来越庞大,形态越来越复杂。除了原有的面向对象的软件、基于组件的软件外,又有了具有多成分、异构性,可动态配置和组合的 Web 应用软件。Internet 的普及以及组件、中间件和 Web Services 等技术的迅速发展和应用使 Web 应用渗透到国计民生的各个领域。越来越多的政府部门和企业都已经或正在开发高性能的集业务流程于一体的 Web 应用,如电子政务、网上购物、网上交易、网络游戏、电子银行等等。

基于模型的软件测试方法与技术^[8-15]在近几年得到了较广泛的关注。Offutt 和 Liu^[16]给出了从 SOFL 形式规格说明产生测试用例的步骤,并通过具体的实例详细说明了测试用例产生的过程。Offutt 等^[17]还给出了一系列基于状态规格说明的测试准则,用来指导基于状态规格说明的系统测试。文献[18]讨论了在模型驱动工程环境下的平台独立的 Web 应用建模与开发。介绍了特殊的元模型(和相关的表示法)用于建模动态 Web 内容。Web 应用被表示成三个独立而相关的模型(业务、超文本、表示模型)。在这些模型中,一种基于 OCL 和 Java 的动作语言用来书写方法和动作,规范约束和表达条件。这些概念在工具 Netsilon 中已得以实现,并且可操作的模型驱动的 Web 信息系统也通过从抽象模型到平台相关模型的转换得到成功的部署。OOWS^[19]是一个模型驱动的 Web 应用建模方法,强调 Web 应用从需求获取到模型建立,经过模型转换到最终实现的开发全过程。研究表明基于模型的测试具有很高的成本效益,一些测试产生策略和模型覆盖准则被开发,一些理论的和商业的工具相继出现^[9]。

基于模型的测试是一种突破性的黑盒测试技术,它以明确描述系统预期行为的抽象模型为依据,根据模型覆盖测试准则自动生成抽象的测试用例,自动地产生测试脚本,执行测试并自动评价测试结果,实现测试过程的完全自动化。总体上说,这方面的研究还处于方法探索阶段,比较实用的工具非常少。这种方法可以使得软件测试和软件设计以及实现同时进行,改变了软件工程中“现在编程,以后测试”的工作方式,使得在整个软件生命期中都可以并行地进行测试工作,从而极大地提高了软件开发的效率。

1.4 Web 应用的测试

Web 应用和 Web 站点没有严格的定义区别,但最基本的一条是,Web 应用可以认为是充当用户接口,以处理服务器端业务逻辑的 Web 站点^[20]。Web 应用已经成为发展最快的一种类型的软件之一^[21]。Web 应用的测试是一项重要而富有挑战性的工作,与传统的软件测试不同,它不但需要检查和验证是否按照设计的要求运行,而且要测试系统在不同用户的浏览器端的显示是否合适,还要从最终用户的角度进行安全性和可用性测试。Web 应用的异构性、动态性、连接的多样性、控制流程的可变性以及需要快速开发与发布等特性给 Web 应用的建模和测试带来了新的挑战。目前还没有比较系统的方法和工具对 Web 应用进行有效的测试,因而 Web 应用的测试重复、低效,导致 Web 应用软件的开发可靠性不高、风险大。本书关注的是 Web 应用的功能测试。

1.4.1 Web 应用的分类与特性

可以从不同的方面对 Web 应用进行分类。根据 Web 应用的发展历史和复杂程度将 Web 应用分为^[20,22]:

- 以文档为中心的 Web 网站(Document Centric)。这是 Web 应用的先驱。Web 页是存储在 Web 服务器中静态的 HTML 文件,根据客户端的请求发出响应信息。Web 页通常使用不同的工具手工更新,这经常会导致信息过时,并带来不一致性问题。这种 Web 网站的主要优点是它的简单性、稳定性以及极快的响应时间,因为 Web