

基于 web 应用程序的自动化测试研究

查满霞

(江苏方天电力技术有限公司, 江苏 南京 211102)

摘 要: Web 应用程序相比于传统的应用程序有其自身的特点, 对软件测试提出了新的要求。该文针对Web 应用程序的特征, 提出了一种Web应用程序的测试框架, 形成了一套规范化、易操作、可扩展的测试流程, 并在该框架的基础上讨论了一种实现技术, 开发出了相应的Web应用程序功能自动化测试的工具。该工具在实践中被证明是有效的, 能及时发现Web 应用程序中的错误, 使迭代开发过程中的回归测试变得简单而方便。

关键词: 自动化测试; 功能测试; 测试框架; web 应用程序

0 引言

随着Internet的普及和电子商务应用的深入, Web 应用程序得到越来越广泛的应用B/S 架构也逐渐战胜C/S 架构成为主流的应用模式, 迭代式开发模式正逐渐取代传统的瀑布开发模式。

与传统的软件相比, Web应用程序具有分布式、并发、多用户异构等特点, 这些特点对测试提出了新的要求。Web测试是保证Web应用程序质量的有效手段, Web测试取得了一些初步研究成果^[1]。目前在Web测试的建模、测试方法和技术以及相关工具等方面的研究已进行了一些尝试^[2]。

软件测试的自动化技术帮助在更短时间内开发出更高质量的产品, 通过代替频繁重复的手工测试从而节省了大量的时间和开支。

1 软件测试自动化概念

首先, 软件自动化测试是一门技术^[3]。对于任何软件系统, 测试者希望通过有限的测试用例发现软件中的大部分缺陷。自动化测试使得所取得的测试用例得以重复测试, 并能保障测试的科学性、严密性、组织性。其次, 自动化测试是一种机制, 它不仅是指运用自动工具进行测试, 而且包括如何管理测试自动化, 如何确定自动测试的方法以及如何组织测试等。

高效的自动测试来源于好的测试软件, 这些测试软件是由经验丰富的测试人员精心设计的, 在此基础上再应用自动化技术就可以实现自动测试。自动测试通常要比手工测试经济得多, 其开销只是手

工测试的一小部分。自动测试的方法越好, 长期使用获得的收益就越大。

2 自动测试的设计

为了更加明确地了解软件测试自动化, 以及自动测试与手工测试的关系, 我们有必要分析一下如何进行自动测试的设计。

在具体讨论之前, 需要明确一个问题——软件测试自动化不等同于软件测试, 从所包含的内容到操作者的所需掌握的技术都有所不同。

2.1 软件测试的步骤

软件测试同其他工作一样, 分为不同的阶段和步骤。人们对软件测试的各个阶段的划分有所不同, 但总的来说可以大致分为以下五个阶段: 标识测试条件、设计测试用例、建立测试用例、执行测试用例、测试结果的比较。对这五个阶段的解释见表 1。

表 1 软件测试的步骤

测试的步骤	内容	特点	重复执行情况
标识测试条件	确定测试的内容和优先级	属于智力活动较多工作	一般只执行一次
设计测试用例	对测试用例进行设计, 确定怎样测试	需要进行创造性的智力活动	一般只执行一次
建立测试用例	建立测试脚本、测试输入、测试数据、期望输出。	使用自动测试比手工测试更复杂	需要重复
执行测试用例	在被测试软件运行时使用测试用例	不需要太多的创造性工作	需要大量重复
测试结果的比较	把测试结果与预期结果进行比较	不需要太多的创造性工作	需要大量重复

一个软件测试应当按照这五个步骤进行。在设计测试用例之前标识测试条件, 建立测试用例之前必须先设计测试用例, 运行测试用例之前必须先建

立测试用例，比较测试结果之前必须先运行测试用例。测试用例应准确地针对被测系统的各项功能进行开发，而且都顺序执行这 5 个测试开发活动。

下面对这五个步骤进行进一步的说明：

(1) 标识测试条件

测试条件 (test condition) 是被测环境的描述，取决于被测试的项目或事件。可以用不同的方法进行描述，如简单的语言描述，表格项形式或类似于流图的图表形式。标识测试条件是指确定测试“什么”，并定义这些测试条件的优先顺序。

(2) 设计测试用例

设计测试用例，确定“怎样”测试。测试用例 (test case) 是按一定顺序执行的与测试目标 (test object)、测试理由或目的相关的一系列测试。测试用例设计包括测试的输入值、期望结果及其与运行测试的有关信息，如环境要求。

(3) 建立测试用例

测试用例包括准备测试脚本、测试输入、测试数据以及期望输出。测试脚本 (test script) 是具有正规语法的数据和指令的集合。测试输入和期望输出可包括在脚本中，也可以是脚本外的一个文件或数据库。测试脚本可以手工也可以不是手工执行。期望输出可以组织成文件形式，用于自动工具。

(4) 执行测试用例

在被测软件运行时使用测试用例。对于手工测试来讲，测试者按事先准备好的手工过程进行测试。测试者输入数据、观察输出、记录发现的问题。对于自动化测试，只需启动测试工具，并告诉工具执行哪些测试用例。测试执行只能在软件开发完成后进行。

(5) 将测试结果与期望输出比较

应该对每次测试的实际输出进行分析研究，判断软件功能是否正确。这种验证可以是非正式的测试者主观判断，也可以是将实际输出与期望输出进行严格准确的比较。

值得一提的是，比较和验证两者是有区别的：工具可能具有比较功能，但不具有验证功能。工具可以将一系列测试结果与另外一些结果比较，但不能断定输出是否正确。验证活动通常要靠测试人员来完成。由测试者来确认或保证比较的测试结果是正确的。因为，如果实际输出与期望输出不一致，可能有多种因素。有可能是软件不正确，也有可能

运行测试的顺序不对，或期望输出的结果不正确，或测试环境设置不正确。而这些因素自动测试是无法准确分析的，需要由人工判断。

2.2 自动测试设计

在软件测试设计的内容中，并不是所有的步骤都适合自动化。前两个测试活动，即标识测试条件和设计测试用例主要为智力活动。最后两个活动，即执行测试用例和比较测试输出相对来说是比较机械的活动。智力活动决定了测试用例的质量，机械活动是体力劳动则适合自动化。同时，前两个活动一般只执行一次，而后两个要多次重复。测试活动是否需要自动化的标准是，如果这个活动是智力因素较少并且重复的机械活动较多则可以自动化，且自动化后的效果较好。因而，执行测试和将测试结果与期望输出比较这两个阶段适合自动测试。(对五个阶段的分析见图 1)

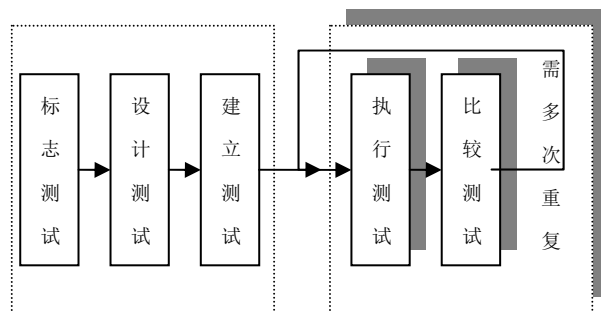


图 1 自动测试设计的分析

由此可见，自动测试并不能占据软件测试的全部内容，需要根据测试的不同阶段具体分析。一个好的测试是把自动测试与手工测试有机的结合，将机械繁琐的测试活动自动化，集中人力对需要较高创造力的阶段进行手工测试。

3 自动测试工具的选择

在进行自动化测试时首要的任务是选择合适的、好的自动测试工具。在工具的选择过程中应该从自己的需求入手，认真分析自身的具体情况，有的放矢地选择测试工具。

3.1 影响选择自动测试工具的因素

选择测试工具需要考虑下面的影响因素：

(1) 测试需求。首先，需要明确测试的需求。在分析测试需求时主要围绕以下几个方面：

1) 自动测试的目标——必须明确想要达到什么目的。

2) 开发和测试过程—保证这些过程自动执行。

3) 自动测试的范围—理解自动测试工具具备的能力。

需要指出的是自动测试的目标应是保证彻底完成回归测试，释放资源，以对一些关键特性进行深层测试，把精力集中在新功能的研究和对新版本的性能测试上。减少测试资源、测试预算和测试队伍决不是自动测试的目标。

(2) 技术质量因素。技术质量因素是指自动测试工具应该解决的测试技术质量问题。这些问题是测试自动化成功的根本依据。例如：

1) 可否解决手工测试的问题，如：时间太长、麻烦、容易出错；

2) 当软件发生小改动时，是否能够及时进行回归测试；

3) 能否解决测试数据或测试用例的设置容易出错的问题；

4) 能够改进测试文件不足的现象，改善测试文档的组织与管理；

5) 是否能够满足测试覆盖的要求，提高测试覆盖率；

6) 可否提高测试效率；

7) 能否与被测试的应用程序更容易更好的结合。

这里我们需要明确一个问题—测试自动化不能把每一个问题都解决的很好。我们在进行分析的时候，要按重要性和对测试工作的价值来分析自己的问题。抓住主要的问题，解决主要矛盾，获得较大利益。

(3) 硬件和软件因素。测试工具就是软件包，因此就可能局限于特定的硬件、软件或操作系统。例如，工作平台、额外的硬件、开发使用的语言等。这些因素对自动测试工具的选择有一定的制约作用。

(3) 经济因素。经济因素通常是选择测试工具的一个重要和直接的原因。

1) 前期调研的费用；

2) 购买测试工具的成本，或者是租借测试工具的成本；

3) 培训使用测试工具的费用；

4) 增加硬件的花费；

5) 增加软件的花费；

6) 维护工具的成本；

7) 其他投入。

其他因素。

3.2 选择测试工具的步骤

测试工具的选择步骤如图 2 所示。

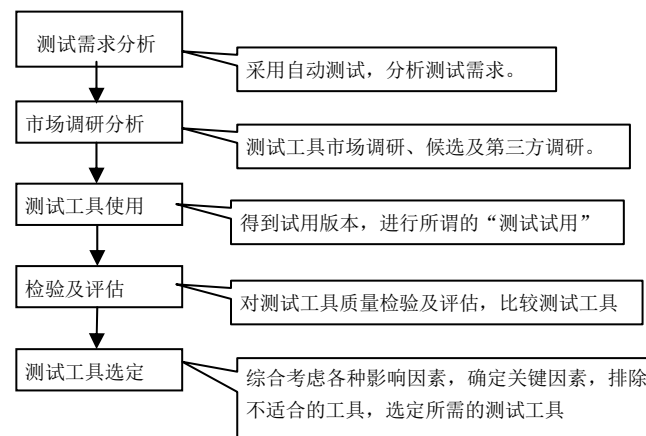


图 2 自动化测试工具的选择步骤

4 自动化测试实例

下面给出自动化回归测试的实例，选自本论文的背景项目电力用户用电信息采集系统。

4.1 电力用户用电信息采集系统

电力用户用电信息采集系统从功能上主要分为：基本应用-档案管理、终端管理、数据采集管理、接口管理；高级应用-计量在线监测、费控管理、有序用电、线损分析、配变监测分析、重点用户监测、数据修复、问题交流平台；运行管理-采集信道管理、时钟管理、SIM 卡管理、采集运维平台；统计查询-综合查询、采集建设情况、数据分析、工单查询、报表管理；系统管理-权限密码管理、系统配置管理、模板管理、信息定制、日志管理、系统使用情况统计、其他功能。电力用户用电信息采集系统的功能结构如图 3 所示。

4.2 自动测试工具 Selenium

4.2.1 Selenium 是什么

Selenium is a suite of tools to automate web browsers across many platforms.这是 selenium 官网关于 selenium 的一段描述。Selenium 是一个强大的开源的 Web 功能自动化测试工具系列。目前主要包括以下 5 部分：

(1) Selenium Core：支持 DHTML 的测试案例（效果类似数据驱动测试），它是 Selenium IDE

- 和 Selenium RC 的引擎。
- (2) Selenium IDE: FireFox 的一个插件，支持脚本录制回放。

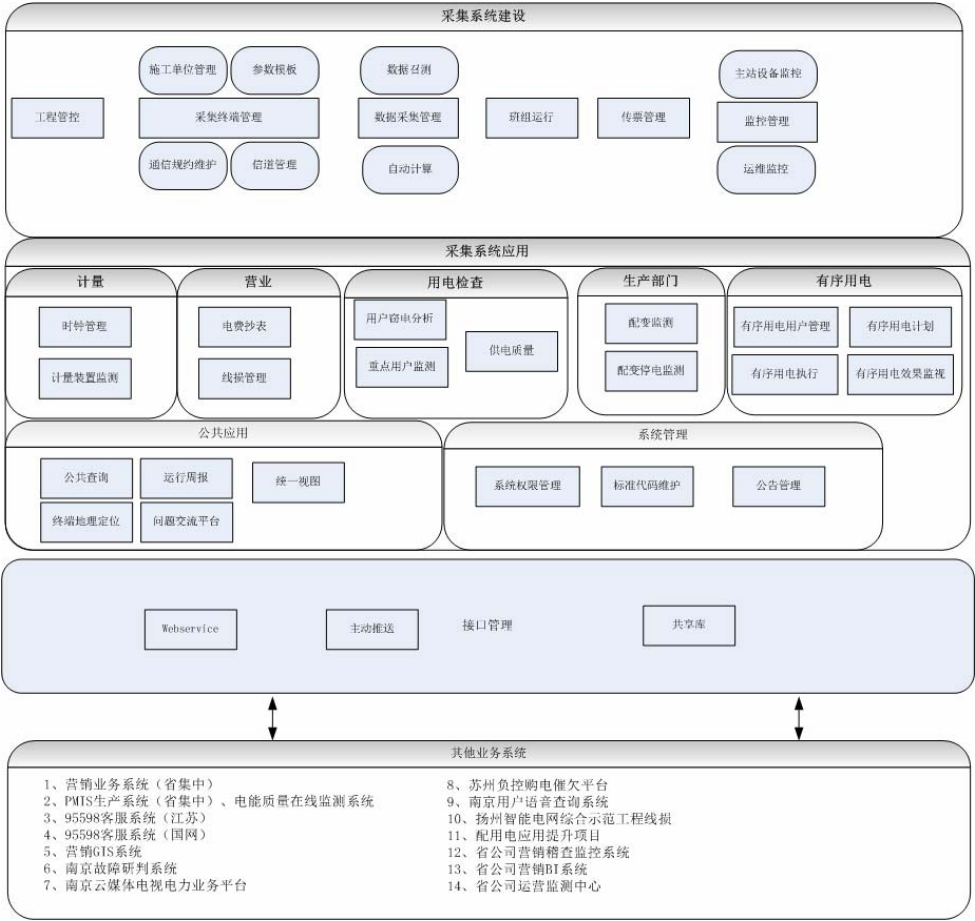
(3) Selenium RC (Remote Control): 是一个客户端/服务器，可以控制本地或其他的 web 浏览器。

(4) Selenium WD: Webdriver，是我们讨论的重点。

(5) Selenium Grid: 允许同时并行地、在不同的环境上运行多个测试任务。
- Selenium 2=Selenium 1 + WebDriver
- 4.2.2 为什么用 Selenium
- (1) Selenium 使用灵活，简单，写出的测试用例非常简洁，优美，易于维护；

(2) 支持多种语言编写用例；

(3) 最主要的还是它是开源的了，非常强大。
- 4.2.3 小趣事 (selenium 的名字的由来)
- Selenium 的中文名字是“硒”，是一种化学元素，它对汞 (Mercury) 有天然的解毒作用，硒是汞的克星。Mercury 的测试工具大家应该都熟悉吧，QuickTest Pro, QC, LR, WR……，他们更能强大，但价格不菲，Selenium 名字的寓意就是让大家远离汞毒。



4.3 自动测试实例

4.3.1 环境准备

先来看一下如何在 Eclipse 下使用 Java 来编写 webdriver 的 case。需要准备的东西: jdk1.6、eclipse、selenium-server-standalone-2.39.0.jar 包。使用浏览器

是 IE8，要注意 IE 的版本是否和 selenium server 的版本匹配。

首先在 eclipse 里新建一个 java project-automation，然后在 automation 项目下找到文件夹 lib，lib 原来放 selenium 的 server 包。如图 4

所示。

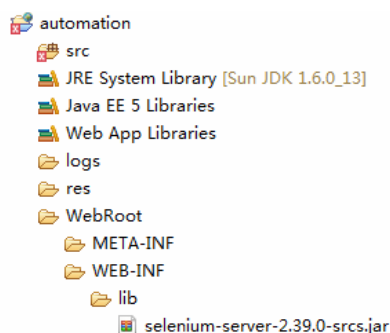


图 4 lib 文件夹

然后将 selenium 加入到 build path 里：选中项目右键→Build Path→Configure Build Path，然后点击 add jars→选我们项目下的 selenium server 就可以了。如图 5 所示。到这里我们的准备就做好了。

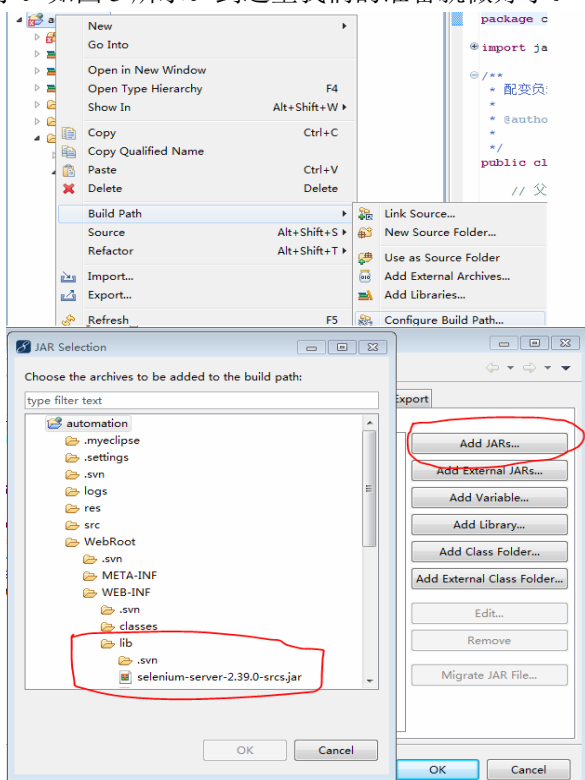


图 5 添加 jar 包

4.3.2 编写脚本

下面新建一个名为 `AdvanceMoreMonitorLoad` 的 java 类，来编写“配变负载率监测”功能的自动化测试脚本，如图 6 所示。

4.3.3 回归测试操作

在 eclipse 上面选择 Run As→Java Application

选项，启动自动化测试程序，打开管理窗口，如图 7。



图 7 自动化测试程序

选择是否是首次运行，按计划运行还是按用例运行，输入用例或计划 ID，点击【运行用例】按钮，就开始执行自动化测试脚本。

4.3.4 测试结果及分析

当对程序进行了修改，或者由于其他原因使程序发生了改动，则对程序重新测试，测试完成可以查看报告以及明细，如图 8 所示。



图 8 查看报告及明细

5 结束语

由上面的自动测试实例，可以看出充分利用自动测试工具确实可以提高测试的效率。

同时，利用测试工具进行自动测试还有一个潜在的好处，使得测试工程师可以有更充裕的时间和精力进行手工测试；无疑这对于测试工作的顺利、高质量的完成，及保障软件的质量都是大有益处的。

参考文献：

- [1] Liu C H.A formal Object-oriented Test Model for Testing Web Applications[D]. Arlington:Graduate School of the University of Texas,2002.
- [2] Xu L,Xu B W,Chen Z Q,et al. Titsworth F Ed.Regression Testing for Web Applications Based on Slicing[C].Proc.of the 27th Annual Int. Computer Software & Applications Conference. Los Alamitos:IEEE Computer Society,2003.
- [3] 郑人杰.计算机软件测试技术[M].北京:清华大学出版社, 1992.

```
package com.frame.script;

import java.util.ArrayList;

/**
 * 配变负载率监测
 *
 * @author pan
 *
 */
public class AdvanceMoreMonitorLoad {

    // 父窗口的查询条件及 第5行td及td的对象 集合
    public static Map<String, Object> newParamsMap = null;

    /**
     * 配变负载率监测查询, 根据日期
     *
     * @param caseId
     *      用例id
     * @return -1: 未执行, 0: 失败, 1: 成功
     */
    public void start(Integer caseId, String planBatchId) {

        // 执行结果标记
        String result = "0";

        // 后台验证及测试
        CaseAndValidateMain cvm = new CaseAndValidateMain();

        /**
         * 执行用例
         */
        LogUtil.logInfo("=====开始[用例执行]=====");
        try {
            // 获取用例id, 获取页面查询条件对应的参数
            Map<String, Object> pageParamsMap = OperateDriver
                .getQueryParams(caseId);

            // 菜单
            OperateDriver.clickMenu("高级应用", "更多", "配变监测", "配变负载率监测");
            OperateDriver.clickMenu("高级应用", "配变监测", "配变负载率监测", "NULL");

            // 配置配变负载率监测页面
            OperateDriver.switchToFrameByID("2833028808iframe");
            OperateDriver.switchToFrameByID("3351395165iframe");

            // 选择周期
            String period = pageParamsMap.get("周期").toString();
            OperateDriver
                .selectDropDownBox(
                    "//input[@id='gridTypeComBox_2ext']/following-sibling::img[1]",
                    period, "周期");

            // 点击日历
            String calendarId = pageParamsMap.get("日期ID").toString();
            String monthYear = pageParamsMap.get("月年").toString();
            String day = pageParamsMap.get("日").toString();

            OperateDriver.clickCalendar(calendarId);
            OperateDriver.selectCalendar(monthYear, day);

            // 查询按钮
            OperateDriver.click("//input[@id='queryBtn']", "查询");

            // 界面上出现的缓冲处理
            OperateDriver.checkBuffer();
            OperateDriver.switchToFrameByID("statReportIframe");

            // 分页总数
            int count = OperateDriver.tableDataCount("dataGrid");

            // 分页数据
            String tableData = OperateDriver
                .tableData(
                    "dataGrid",
                    "1=.",
                    "Name=查询列表^BeginRow=4^KeyCol=1^TagCol=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20");

            // 分页总数的期望的结果
            Map<String, Object> paramsMap = new HashMap<String, Object>();
            paramsMap.put("keyword_value", count);
            paramsMap.put("case_id", caseId);
            paramsMap.put("keyword_name",
                "AdvanceMoreMonitorLoad-CurrentMonth-Count");
            paramsMap.put("statementId",
                "ValidateParam.updateCasedataByCaseIdAndKeyname");
            cvm.updateCasedataByCaseIdAndKeyname(paramsMap);

            // 分页数据的期望结果
            paramsMap.clear();
            paramsMap.put("keyword_value", tableData);
            paramsMap.put("case_id", caseId);
            paramsMap
                .put("keyword_name", "AdvanceMoreMonitorLoad-CurrentMonth");
            paramsMap.put("statementId",
                "ValidateParam.updateCasedataByCaseIdAndKeyname");
            cvm.updateCasedataByCaseIdAndKeyname(paramsMap);

            // 执行成功
            result = "1";

        } catch (Exception e) {
            e.printStackTrace();
        }
        LogUtil.logInfo("=====结束[用例执行]=====");

        // 添加日志及后台验证
        cvm.caseAndValidateAndRecordLogsRun(caseId, planBatchId, result);
    }

}
```

图 6 配变负载率监测功能自动化测试脚本

作者简介:

查满霞(1983-), 女, 江西九江人, 工程师, 从事电力行业
软件研发工作。