

自动化测试概述


方案撰写：许玉波

版本： V1.0

撰写时间： 20140915

个人简介

- 10年软件测试经验;
- 有多少年QTP,Loadrunner,Jmeter性能测试项目经验;
- 精通自动化测试框架;
- 曾为某大公司提供过自动化测试解决方案, 脚本开发;
- 曾就职于某大型外资企业, 任企业测试技术部经理, 负责公司自动化测试框架搭建, 技术培训, 技术研究

- 
- 理解自动化测试
 - 自动化测试工具

理解自动化测试

- 什么是自动化测试
- 手工测试的局限性
- 自动化测试的优点
- 自动化测试认识的误区
- 什么情况适合自动化测试
- 什么情况不适合自动化测试

什么是自动化测试

概念：

通过对工具的使用，增加或减少了手工或人为参与或干预非技巧性、重复或冗长工作。

自动化测试就是希望能够通过自动化测试工具或其他手段，按照测试工程师的预定计划进行自动的测试，目的是减轻手工测试的劳动量，从而达到提高软件质量的目的。自动化测试的目的在于发现老缺陷和提示新缺陷。而手工测试的目的在于发现新缺陷。

自动化测试的意义

- 通常，软件测试的工作量很大（据统计，测试会占用到40%的开发时间；一些可靠性要求非常高的软件，测试时间甚至占到开发时间的60%）。而测试中的许多操作是重复性的、非智力性的和非创造性的，并要求做准确细致的工作，计算机就最适合于代替人工去完成这样的任务。
- 软件自动化测试是相对手工测试而存在的，主要是通过所开发的软件测试工具、脚本等来实现，具有良好的可操作性、可重复性和高效率等特点。

手工测试的局限性

- 通过手工测试无法做到覆盖所有代码路径。
- 简单的功能性测试用例在每一轮测试中都不能少，而且具有一定的机械性、重复性，工作量往往较大。
- 许多与时序、死锁、资源冲突、多线程等有关的错误，通过手工测试很难捕捉到。
- 进行系统负载、性能测试时，需要模拟大量数据或大量并发用户等各种应用场合时，很难通过手工测试来进行。
- 进行系统可靠性测试时，需要模拟系统运行10年、几十年，以验证系统能否稳定运行，这也是手工测试无法模拟的。
- 如果有大量（几千）的测试用例，需要在短时间内（1天）完成，手工测试几乎不可能做到。

自动化测试的优点

- **对新版本执行快速的回归测试。**

对于产品型的软件，每发布一个新的版本，其中大部分功能和界面都和上一个版本相似或完全相同，这部分功能特别适合于自动化测试，从而可以让测试达到测试每个特征的目的

- **执行更多更频繁的测试**

对于产品型软件或需求不断更新的系统，每一版产品发布或系统更新的周期就只有短短的几个月，这就意味着开发周期也只有短短的数月，而在测试期间是每天或每几天要发布一个版本供测试人员测试，一个系统的功能点少则上百多则上千上万，使用手工测试是非常耗时和繁琐的，这样频繁的重复劳动必然会导致测试人员产生厌倦心理、工作效率低下

- **弥补手工测试难实现的不足**

压力测试、并发测试、大数据量测试、崩溃性测试等，都需要成百上千的用户同时对系统加压才能实现其效果，用人来测试是不可能达到的，也是不现实的。在没有引入自动化测试工具之前，为了测试并发，组织几十号人在测试经理的口令：1—、2—、3！下，同时按下同一个按钮，但如果需要更大的并发量，就很难实现了。

自动化测试的优点

- **增强测试的稳定性和可靠性。**

由于每次自动化测试运行的脚本是相同的，所以每次执行的测试具有一致性，而这一点手工测试是很难做到的。由于自动化测试的一致性，很容易发现被测软件的任何改变。

- **更好的利用资源**

理想的自动化测试能够按计划完全自动的运行，测试人员可以设置自动化测试程序在周末和晚上执行测试，白天上班的时候测试人员就可以收集测试所发现的缺陷，并交给开发人员修改，同时测试人员可以在白天开发新增功能的自动化测试脚本，或对已有的脚本不适合的地方进行修改。这样充分的利用了公司的资源，也避免了开发和测试之间的等待。

- **解决测试与开发之间的矛盾**

通常在开发的末期，进入集成测试阶段，由于每发布一个版本的初期，测试系统的错误比较少，这时开发人员有等待测试人员测试出错误的时间。事实上在叠代周期很短的开发模式中，存在更多的矛盾，但自动化测试可以解决其中的主要矛盾。

- **快速实现多平台的测试**

总之软件测试实行自动化进程，绝不是因为厌烦了重复的测试工作，而是因为测试工作的需要，更准确地说是回归测试和系统测试的需要。

自动化测试认识的误区

自动化测试好处很多，但也有很多的局限，正因为一些人只认识到了自动化测试的优点，导致对它的期望太高，所以产生了很多执行自动化测试失败的例子

- **期望自动化测试能取代手工测试**

某些情况下自动化测试并不适用，所以不能期望在所有的情况下都使用自动化测试来取代手工测试，测试主要还是要靠人工的。

- **期望自动测试发现大量新缺陷**

自动化测试只能发现已知的问题和提示问题，所以不能期望自动化测试去发现更多新的缺陷，事实证明新缺陷越多，自动化测试失败的几率就越大。发现更多的新缺陷应该是手工测试的主要目的。

- **工具本身不具有想象力**

对于一些界面美观和易用性方面的测试，自动化测试工具无能为力

自动化测试认识的误区

- 只要使用自动化测试，就能缩短测试时间、提供测试效率

自动化测试的前期实现要花费更多的时间，相比创建和执行一个手工测试用例，要花费3~10倍的时间来开发、验证和文档化一个自动化测试用例

- 自动化测试工具使用了图形化界面，很容易上手，对人员的要求不高

简单的“录制 / 回放”方法并不能实现有效的、长期的自动化测试，测试人员还需要对脚本进行优化，这就需要测试人员具有设计、开发、测试、调试和编写代码的能力，最理想的候选人是既有编程经验，又有测试经验。测试过程中还需要安排专业人员对测试脚本库中的脚本进行维护

什么情况适合自动化测试

- 产品型项目

产品型的项目，每个项目只改进少量的功能，但每个项目必须反反复复的测试那些没有改动过的功能。这部分测试完全可以让自动化测试来承担，同时可以把新加入的功能的测试也慢慢地加入到自动化测试当中

- 增量开发、持续集成的项目

由于这种开发模式是频繁的发布新版本进行测试，也就需要频繁的自动化测试，以便把人从中解脱出来测试新的功能。

- 回归测试

回归测试是自动化测试的强项，它能够很好的验证你是否引入了新的缺陷，老的缺陷是否修改过来了。在某种程度上可以把自动化测试工具叫做回归测试工具

什么情况适合自动化测试

- **多次重复、机械性操作**

自动化测试最适用于多次重复、机械性动作，这样的测试对它来说从不会失败。比如要向系统输入大量的相似数据来测试

- **需要频繁运行测试**

在一个项目中需要频繁的运行测试，测试周期按天算，就能最大限度的利用测试脚本，提高工作效率

- **性能、压力测试**

实现多人同时对系统进行操作时是否正常处理和响应以及系统可承受的最大访问量的测试

什么情况不适合自动化测试

自动化测试不是适合所有公司、所有项目

- **定制型项目（一次性的）**

为客户定制的项目，维护期由客户方承担的，甚至采用的开发语言、运行环境也是客户特别要求的，即公司在这方面的测试积累就少，这样的项目不适合作自动化测试

- **项目周期很短的项目**

项目周期很短，测试周期很短，就不值得花精力去投资自动化测试，好不容易建立起的测试脚本，不能得到重复的利用是不现实的。

- **业务规则复杂的对象**

业务规则复杂的对象，有很多的逻辑关系、运算关系，工具就很难测试

什么情况不适合自动化测试

- **美观、声音、易用性测试**

人的感观方面的：界面的美观、声音的体验、易用性的测试，也只有人来测试

- **测试很少运行**

测试很少运行，对自动化测试就是一种浪费。自动化测试就是让它不厌其烦的、反反复复的运行才有效率

- **软件不稳定**

软件不稳定，则会由于这些不稳定因素导致自动化测试失败。只有当软件达到相对的稳定，没有界面性严重错误和中断错误才能开始自动化测试

- **涉及物理交互**

工具很难完成与物理设备的交互，比如刷卡的测试等

- 自动化测试工具的特征
- 自动化测试工具的分类
- 选择自动化测试工具
- 常用测试工具概要

自动化测试工具的特征

- 支持脚本化语言 (Scripting Language)
- 对程序界面中对象的识别能力
- 支持函数的可重用
- 支持外部函数库
- 抽象层—将程序界面中的对象实体映射成逻辑对象
- 分布式测试 (Distributed Test) 的支持
- 支持数据驱动测试 (Data-Driven Test)
- 错误处理
- 调试器 (Debugger)
- 源代码管理
- 支持脚本的命令行 (Command Line) 方式



自动化测试工具的分类

测试工具可以从两个不同的方面去分类。

- 根据测试方法不同，自动化测试工具可以分为：

白盒测试工具、黑盒测试工具

- 根据测试的对象和目的，自动化测试工具可以分为：

单元测试工具、功能测试工具、负载测试工具、性能测试工具、Web测试工具、数据库测试工具、回归测试工具、嵌入式测试工具、页面链接测试工具、测试设计与开发工具、测试执行和评估工具、测试管理工具等

白盒测试工具

- 白盒测试工具一般是针对被测源程序进行的测试，测试所发现的故障可以定位到代码级。根据测试工具工作原理的不同，白盒测试的自动化工具可分为静态测试工具和动态测试工具。
 - 静态测试工具——是在不执行程序的情况下，分析软件的特性。静态分析主要集中在需求文档、设计文档以及程序结构方面。按照完成的职能不同，静态测试工具包括以下几种类型：
 - (1) 代码审查 (2) 一致性检查 (3) 错误检查 (4) 接口分析
 - (5) 输入输出规格说明分析检查 (6) 数据流分析
 - (7) 类型分析 (8) 单元分析 (9) 复杂度分析

白盒测试工具

- 动态测试工具——是直接执行被测程序以提供测试活动。它需要实际运行被测系统，并设置断点，向代码生成的可执行文件中插入一些监测代码，掌握断点这一时刻程序运行数据（对象属性、变量的值等），具有功能确认、接口测试、覆盖率分析、性能分析等性能。动态测试工具可以分为以下几种类型：
 - (1) 功能确认与接口测试
 - (2) 覆盖测试
 - (3) 性能测试
 - (4) 内存分析
- 常用的动态工具有：
 - Compuware公司的DevPartner
 - IBM公司的 Rational

黑盒测试工具

- 黑盒测试工具是在明确软件产品应具有的功能的条件下，完全不考虑被测程序的内部结构和内部特性，通过测试来检验软件功能是否按照软件需求规格的说明正常工作。
- 按照完成的职能不同，黑盒测试工具可以分为：
 - 功能测试工具——用于检测程序能否达到预期的功能要求并正常运行。
 - 性能测试工具——用于确定软件和系统的性能。
- 常用的黑盒测试工具有：
 - Compuware公司的QACenter
 - IBM公司的Rational TeamTest

测试设计与开发工具

- 测试设计是说明被测软件特征或特征组合的方法，并确定选择相关测试用例的过程。
- 测试开发是将测试设计转换成具体的测试用例的过程。
- 测试设计和开发需要的工具类型有：
 - 测试数据生成器
 - 基于需求的测试设计工具
 - 捕获/回放
 - 覆盖分析

测试执行和评估工具

- 测试执行和评估是执行测试用例并对测试结果进行评估的过程，包括选择用于执行的测试用例、设置测试环境、运行所选择的测试用例、记录测试执行过程、分析潜在的故障，并检查测试工作的有效性。
- 评估类工具对执行测试用例和评估测试结果过程起到辅助作用。
- 测试执行和评估类工具有：
 - 捕获/回放
 - 覆盖分析
 - 存储器测试

测试管理工具

- 测试管理工具用于对测试过程进行管理，帮助完成制定测试计划，跟踪测试运行结果。通常，测试管理工具对测试计划、测试用例、测试实施进行管理，还包括缺陷跟踪管理等。
- 常用的测试管理工具有：
 - IBM公司的Rational Test Manager
 - HP公司的ALM
- 测试管理工具包括以下内容：
 - 测试用例管理
 - 缺陷跟踪管理（问题跟踪管理）
 - 配置管理



选择自动化测试工具

- 测试人员在选择和使用自动化测试工具时，可以从以下角度来考虑：
 - 按照用途选择匹配的测试工具
 - 在适当的生命周期选择测试工具
 - 按照测试人员的实际技能选择匹配的测试工具
 - 选择一个可提供的测试工具



常用测试工具

- 目前，软件测试方面的工具很多，主要有MercuryInteractive（MI）、Rational、Compuware、Segue和Empirix等公司的产品，而MI公司和Rational公司的产品占了主流。
- 开源测试工具
Appium, TestLink, Selenium, RFS

HP公司测试工具

- HP质量中心：提供一个全面的、基于Web的集成系统，可在广泛的应用环境下自动执行软件质量管理和测试。其主要产品如下：
 - QTP：是一种企业级的用于检验应用程序是否如期运行的功能性测试工具。通过自动捕获，检测，和重复用户交互的操作，QTP 能够辨认缺陷并且确保那些跨越多个应用程序和数据库的业务流程在初次发布就能避免出现故障，并且保持长期可靠运行。
 - Loadrunner：是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner 能够对整个企业架构进行测试。通过使用 LoadRunner ，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。

HP公司测试工具

- **ALM:** 是基于Web的测试管理解决方案，它可以在公司内部进行全球范围的测试协调。ALM能够在一独立的应用系统中提供需求管理功能，并且可以把测试需求管理于测试计划、测试日程控制、测试执行和错误跟踪等功能融合为一体，因此极大地加速了测试的进程。QC提供完整且无限制的测试管理框架，实现对应用测试全部阶段的管理与控制。
- **UFT:** 是一个功能测试自动化工具，主要应用在回归测试中。QuickTest针对的是GUI应用程序，包括传统的Windows应用程序，以及现在越来越流行的Web应用。它可以覆盖绝大多数的软件开发技术，简单高效，并具备测试用例可重用的特点。其中包括：创建测试、插入检查点、检验数据、增强测试、运行测试、分析结果和维护测试等方面。

Rational公司测试工具

- Rational (<http://www-900.ibm.com/cn/software/rational/>) 公司产品如下：
 - Rational Functional Tester: 对 Java、Web 和基于 VS.NET WinForm 的应用程序进行高级自动化功能测试。
 - Rational Functional Tester Extension for Terminal-based Applications: 扩展了 Rational Functional Tester, 以支持基于终端的应用程序的测试。
 - Rational Manual Tester: 使用新测试设计技术来改进人工测试设计和执行工作。
 - Rational Performance Tester: 检查可变多用户负载下可接受的应用程序响应时间和可伸缩性。
 - Rational Purify for Linux and UNIX: 为 Linux 和 UNIX 提供了内存泄漏和内存损坏检测。
 - Rational Purify for Windows: 为 Windows 提供了内存泄漏和内存损坏检测。

Rational公司测试工具

- Rational PurifyPlus 企业版：为 Windows、Linux 和 UNIX 提供了运行时分析。
- Rational PurifyPlus for Linux and UNIX：为 基于 Linux 和 Unix 的 Java 和 C/C++ 开发提供了分析工具集。
- Rational PurifyPlus for Windows：为基于 Windows的Java、C/C++、Visual Basic 和 托管 .NET 开发提供了运行时分析。
- Rational Robot：客户机/服务器应用程序的通用测试自动化工具。可以对使用各种集成开发环境（IDE）和语言建立的软件应用程序，创建、修改并执行自动化的功能测试、分布式功能测试、回归测试和集成测试。
- Rational TestManager：提供开放、可扩展的测试管理。
- Rational Test RealTime：支持嵌入式和实时的跨平台软件的组件测试和运行时分析。

Compuware公司测试工具

- Compuware (<http://www.compuware.com>) 公司的QACenter家族集成了一些强大的自动工具，这些工具符合大型机应用的测试要求，使开发组获得一致而可靠的应用性能。QACenter帮助所有的测试人员创建一个快速，可重用的测试过程。这些测试工具自动帮助管理测试过程，快速分析和调试程序，包括针对回归，强度，单元，并发，集成，移植，容量和负载建立测试用例，自动执行测试和产生文档结果。QACenter主要包括以下几个模块：
 - QARun：应用的功能测试工具。
 - QALoad：强负载下应用的性能测试工具。
 - QADirector：测试的组织设计和创建以及管理工具。
 - TrackRecord：集成的缺陷跟踪管理工具。
 - EcoTools：高层次的性能监测工具。

其他公司测试工具

- segue 公司的SilkTest (<http://www.segue.com>)：是业界领先的、用于对企业级应用进行功能测试的产品，可用于测试Web、Java或是传统的C/S结构。SilkTest提供了许多功能，使用户能够高效率地进行软件自动化测试。这些功能包括：测试的计划和管理；直接的数据库访问及校验；灵活、强大的4Test脚本语言，内置的恢复系统(Recovery System)；以及具有使用同一套脚本进行跨平台、跨浏览器和技术进行测试的能力。

其他公司测试工具

- AdventNet公司的 QEngine (<http://www.adventnet.com>)：是一个应用广泛且独立于平台的自动化软件测试工具，可用于Web功能测试、web性能测试、Java应用功能测试、Java API测试、SOAP测试、回归测试和Java应用性能测试。支持对于使用HTML、JSP、ASP、.NET、PHP、JavaScript/VBScript、XML、SOAP、WSDL、e-commerce、传统客户端/服务器等开发的应用程序进行测试。此工具以Java开发，因此便于移植和提供多平台支持。

其他公司测试工具

- Radview公司的TestView系列Web性能测试工具和WebLoad Analyzer性能分析工具，旨在测试Web应用和Web服务的功能、性能、程序漏洞、兼容性、稳定性和抗攻击性，并且能够在测试的同时分析问题原因和定位故障点。整套Web性能测试和分析工具包含两个相对独立的子系统：Web性能测试子系统、Web性能分析子系统。其中Web性能测试子系统包含3个模块：TestView Manager、WebFT以及WebLoad。Web性能分析子系统只有WebLoad Analyzer。

其他公司测试工具

- 美国IXIA公司的应用层性能测试软件IxChariot是一个独特的测试工具，也是在应用层性能测试领域得到业界认可的测试系统。对于企业网而言，IxChariot可应用于设备选型、网络建设及验收、日常维护等3个阶段，提供设备网络性能评估、故障定位和SLA基准等服务。
- IxChariot由两部分组成：控制端（Console）和远端（Endpoint），两者都可安装在普通PC或者服务器上，控制端安装在Windows操作系统上，远端支持各种主流的操作系统。控制端为该产品的核心部分，控制界面（也可采用命令行方式）、测试设计界面、脚本选择及编制、结果显示、报告生成以及API接口提供等都由控制端提供。远端根据实际测试的需要，安装在分布的网络中，负责从控制端接收指令、完成测试并将测试数据上报到控制端。

一些开源测试工具

- 功能测试工具

- **Linux Test Project** (<http://ltp.sourceforge.net/>) : Linux Test Project是一个测试Linux内核和内核相关特性的工具集合。该工具的目的在于通过把测试自动化引入到Linux内核测试, 提高Linux的内核质量。

使用环境: Linux

- **MaxQ** (<http://maxq.tigris.org/>) : MaxQ是一个免费的功能测试工具。它包括一个HTTP代理工具, 可以录制测试脚本, 并提供回放测试过程的命令行工具。测试结果的统计图表类似于商用测试工具, 比如Astra QuickTest和Empirix e-Test, 这些商用工具都很昂贵。MaxQ希望能够提供一些关键的功能, 比如HTTP测试录制回放功能, 并支持脚本。

使用环境: Java 1.2以上版本

- **WebInject** (<http://www.webinject.org/>) : WebInject是一个针对Web应用程序和服务的免费测试工具。它可以通过HTTP接口测试任意一个单独的系统组件。可以作为测试框架管理功能自动化测试和回归自动化测试的测试套。

使用环境: Windows, OS Independent, Linux

一些开源测试工具

- 单元测试工具

JUNIT (CppUnit) : JUnit是一个开源的java测试框架，它是Xuint测试体系架构的一种实现。在JUnit单元测试框架的设计时，设定了三个总体目标，第一个是简化测试的编写，这种简化包括测试框架的学习和实际测试单元的编写；第二个是使测试单元保持持久性；第三个则是可以利用既有的测试来编写相关的测试。

使用环境：Windows, OS Independent, Linux

一些开源测试工具

- 性能测试工具

- Apache JMeter (<http://jakarta.apache.org/jmeter/>) :
Apache JMeter是100%的Java桌面应用程序，它被设计用来加载被测试软件功能特性、度量被测试软件的性能。设计Jmeter的初衷是测试Web应用，后来又扩充了其它的功能。Jmeter可以完成针对静态资源和动态资源（讹误女监，Servlets, Perl脚本，Java对象，数据查询s, FTP服务等）的性能测试。。 Jmeter可以模拟大量的服务器负载、网络负载、软件对象负载，通过不同的加载类型全面测试软件的性能。Jmeter提供图形化的性能分析。

使用环境：Solaris, Linux, Windows (98, NT, 2000). JDK1.4以上.

- DBMonster (<http://dbmonster.kernelpanic.pl/>) : DBMonster是一个生成随机数据，用来测试SQL数据库的压力测试工具。

使用环境：OS Independent

一些开源测试工具

- OpenSTA (Open System Testing Architecture)
(<http://portal.opensta.org/index.php>) : 基于CORBA的分布式软件测试构架。使用OpenSTA, 测试人员可以模拟大量的虚拟用户。OpenSTA的结果分析包括虚拟用户响应时间、web服务器的资源使用情况、数据库服务器的使用情况, 可以精确的度量负载测试的结果。

使用环境: OS Independent

- TPTEST (<http://tpptest.sourceforge.net/about.php>) : 工具描述: TPTest的提供测试Internet连接速度的简单方法。

使用环境: MacOS/Carbon、 Win32

- Web Application Load Simulator
(<http://www.openware.org/loadsim/index.html>) : LoadSim是一个网络应用程序的负载模拟器。

使用环境: JDK 1.3以上

一些开源测试工具

- 缺陷管理工具

- **Mantis** (<http://mantisbt.sourceforge.net/>) : Mantis是一款基于WEB的软件缺陷管理工具, 配置和使用都很简单, 适合中小型软件开发团队。

使用环境: MySQL, PHP

- **Bugzilla** (<http://www.mozilla.org/projects/bugzilla/>) : 一款软件缺陷管理工具。

使用环境: TBC

一些开源测试工具

- 测试管理工具

- **TestLink** (<http://testlink.sourceforge.net/docs/testLink.php>) : 基于WEB的测试管理和执行系统。测试小组在系统中可以创建、管理、执行、跟踪测试用例，并且提供在测试计划中安排测试用例的方法。

使用环境：Apache, MySQL, PHP

- **Bugzilla Test Runner** (<http://sourceforge.net/projects/testrunner/>) : Bugzilla Test Runner基于Bugzilla缺陷管理系统的测试用例管理系统。

使用环境：Bugzilla 2.16.3 or above (bugzilla是一个可以发布bug以及跟踪报告bug进展情况的开源软件)

