

如何保证软件测试的质量

随着软件测试受关注程度越来越高，如何采用技术手段有效提高软件测试质量就成了软件测试领域的一个重要课题。本文从软件测试的基本概念开始，对如何以软件测试性设计为中心、合理运用软件测试技术来提升软件测试质量提出了自己的看法。软件产品的质量取决于软件开发过程，软件测试作为软件生存期中的一个重要阶段，受重视程度越来越高。软件测试是保证软件质量和可靠性的关键步骤，也是用来验证软件是否能够完成所期望功能的唯一有效的方法。测试已不仅仅局限于软件开发中的一个阶段，它已开始贯穿整个软件开发过程，进行测试的时间越早，整个软件开发成本下降就越多。

随着软件测试的地位逐步提高，软件测试技术也随之快速发展，逐渐从过去手工式的测试向测试自动化方向发展，而测试自动化则能更好的提高软件测试质量。目前，软件测试自动化的研究领域主要集中在软件测试流程的自动化管理以及动态测试的自动化(如单元测试、功能测试以及性能测试方面)。在这两个领域，与手工测试相比，测试自动化的优势是明显的。首先，自动化测试可以提高测试效率，使测试人员更加专注于新的测试模块的建立和开发，从而提高测试覆盖率；其次，自动化测试更便于测试资产的数字化管理，使得测试资产在整个测试生命周期内可以得到复用，这个特点在功能测试和回归测试中尤其具有意义；此外，测试流程自动化管理可以使机构的测试活动开展更加过程化，这很符合 CMMI 过程改进的思想。根据 Oppenheimer Funds 的调查，在 2001 年前后的 3 年中，全球范围内

由于采用了测试自动化手段所实现的投资回报率高达 1500%。

可以说，实施测试自动化是软件行业一个不可逆转的趋势，如果在这个领域走在了前列，无论从企业的核心竞争力还是个人的工作技能来说，都有巨大的优越性。

自动化测试是把以人为驱动测试行为转化为机器执行的一种过程。通常，在设计了测试用例并通过评审之后，由测试人员根据测试用例中描述的规程一步步执行测试，得到实际结果与期望结果的比较。在此过程中，为了节省人力、时间或硬件资源，提高测试效率，便引入了自动化测试的概念。相对于手工测试而言，测试自动化通常具有速度快、执行效率高、测试全面，可连续工作，执行过程受外界因素干扰小、测试结果准确等优点。缺点是前期投入较大，所以在采用测试自动化之前需做好相应的评估工作。本文以 Selenium 自动化测试框架为应用背景，对测试自动化在系统测试过程中的应用做了初步的研究。

接下来我们就具体谈谈如何保证软件测试的质量。

(1) 选择合适的自动化测试方案测试自动化的引进和实施，不仅涉及测试工作本身流程上、组织结构上的调整与改进，甚至也包括需求、设计、开发、维护及配置管理等其他方面的配合。如果对这些必要的因素没有考虑周全的话，必然在实施过程中处处碰壁，既定的实施方案也无法开展。

(2) 尽管自动化测试可以降低人工测试的工作量，但并不能完全取代手工测试。100% 的自动化测试只是一个理想目标，即便一些如

SAP、Oracle ERP 等测试库规划十分完善的套件，其测试自动化率也不会超过 70%。所以一味追求测试自动化只会给企业带来运作成本的急剧上升。再次，实施测试自动化需要企业有相对规模的投入，对企业运作来说，投入回报率将是决定是否实施软件测试自动化的最终指挥棒，因此企业在决定实施软件测试自动化之前，必须要做量化的投资回报分析。

(3) 实施软件测试自动化并不意味着必须采购强大的自动化软件测试工具或自动化管理平台，在企业内部通常存在许多不同种类的应用平台，应用开发技术也不尽相同，甚至在一个应用中可能就跨越了多种平台，或同一应用的不同版本之间存在技术差异。所以选择软件测试自动化方案必须深刻理解这一选择可能带来的变动、来自诸多方面的风险和成本开销。毕竟软件质量的保证不是依靠产品或技术，更多的因素在于高素质的人员和合理有效的流程。

2、选型分析

实施自动化测试之前需要对软件开发过程进行分析，以观察其是否适合使用自动化测试。通常需要同时满足以下条件：

1) 软件需求变动不频繁

测试脚本的稳定性决定了自动化测试的维护成本。如果软件需求变动过于频繁，测试人员需要根据变动的需求来更新测试用例以及相关的测试脚本，而脚本的维护本身就是一个代码开发的过程，需要修改、调试，必要的时候还要修改自动化测试的框架，如果所花费的成本不低于利用其节省的测试成本，那么自动化测试便是失败的。项目

中的某些模块相对稳定，而某些模块需求变动性很大，此时可对相对稳定的模块进行自动化测试，而变动较大的仍是用手工测试。

2) 项目周期足够长

由于自动化测试需求的确定、自动化测试框架的设计、测试脚本的编写与调试均需要相当长的时间来完成。这样的过程本身就是一个测试软件的开发过程，需要较长的时间来完成。如果项目的周期比较短，没有足够的时间去支持这样一个过程，那么实施自动化测试就变成空谈。

基于上述条件，我们在实际应用中多选取在项目规模较大的回归测试环节，或者增量式开发、持续集成的项目中采用自动化测试。

3、测试框架评估

自动化测试框架从最初简单的脚本录制/回放发展到结构化脚本、数据驱动(data •driven)脚本，再发展到关键字驱动(keyword. driven)脚本，以及相继出现的各种自动化测试框架，逐渐形成了良好的脚本开发环境或平台，使得自动化测试脚本的开发更具开放性、可视性和层次性，测试人员开发和维护脚本都变得更轻松、容易，从而在整体上进一步提高自动化测试的效率和应用范围。

现有可提供自动化测试解决方案的产品很多，但是其应用结果往往不竟如人意，主要存在：定位控件不方便；验证数据不方便；代码维护不方便等。对于经常涉及手工任务操作的 Web 应用程序测试，为了尽可能地消除人为因素，节约手工操作的人员成本和时间，实现跨平台的 Web 测试框架，我们发现了 Selenium 这个开源的软件自动化测

试框架。

a) 自动化框架的选择

测试框架的选择是成功实施测试自动化的基础，必须考虑各种因素以使测试自动化更加有效地被使用。Selenium 是一种 Web 应用的自动测试工具，通过模拟用户对 Web 页面的各种操作，可以精确重现软件测试人员编写的 Test Cases 步骤。相对于一般的脚本录制自动化框架，其优势在于：

(1) Selenium 测试直接在浏览器中运行，就像真实用户所做的一样。Selenium 测试可以在 Windows、Linux 和 Macintosh 上的 Internet Explorer、Mozilla 和 Firefox 中运行。其他测试工具都不能覆盖如此多的平台。

(2) Selenium 提供了组件化的测试用例开发方法，建立了灵活的表达方式，很大程度地提高了运行脚本的复用率，节省了脚本开发时间。

(3) 通过 Selenium 开发的自动测试框架，对组件进行了有效封装，大大降低了维护工作量。不会产生传统自动化测试工具录制回放方法中出现的随程序变更，需要重复录制脚本的问题。

(4) Selenium 以业务为驱动，可以通过测试数据流的配置，快速、灵活地实现不同的业务流分支的回归测试。

(5) Selenium 的测试用例、测试结果数据统一在配置库中管理，实现了测试工具无关性。可以很好的进行移植，并为用户提供了一套完整的测试资产库。

但是 Selenium 是轻量的测试框架，脚本所处理的测试用例构成简单，其实质就是通过 HTTP 协议，发送请求(request)来完成测试用例，所以很困难处理业务逻辑关系强的测试用例。

b) 选择合适的测试用例

大部分自动化测试项目失败的原因主要归咎于被测试应用程序的快速变化、不恰当的测试用例、不可靠的框架、脚本编程等问题。并非所有的测试用例都可以用自动化来完成，因此需要对用例进行挑选，选择合适的用例作为自动化测试用例。自动化测试的成本是巨大的，一般来说，一个脚本运行 6—7 次才算收回成本，因此不可寄予自动化测试过高期望。通常需要结合测试用例的复杂度的评估来考虑选择的测试用例以及个数。这样会带来较低的维护成本，实现更重要的业务价值。

首先把测试用例按一定的原则分为简单、中等、复杂 3 大类。然后从这 3 大类的测试用例中按一定的比例来抽取需要实现自动化的用例。测试用例的复杂度分组可以通过综合分析测试用例包含的操作步骤，以及测试用例所包含的检查点个数来判定。

总之，为了保证软件测试的质量要做到：

1. 选择合适的自动化测试方案测试自动化的引进和实施。
2. 结合使用自动化测试和手动测试。
3. 提高测试人员的素质和完善测试流程。