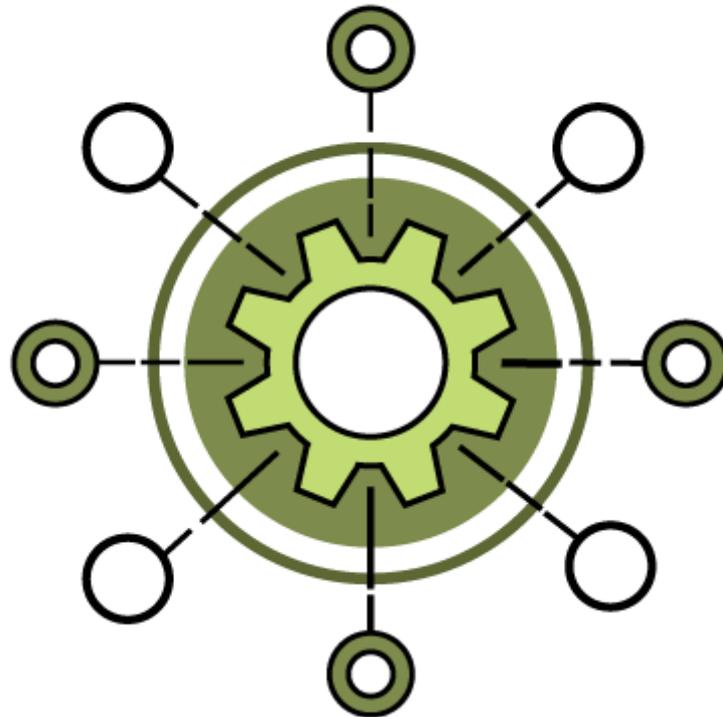


自动化测试方法、流程与工具

来源	来自于 JavaTpoint 网站
翻译	Aaron (火龙果软件)
说明	本文将介绍自动化测试简介，为什么我们要执行自动化测试，自动化测试中使用的不同方法，自动化测试流程以及面临哪些不同的挑战，最后介绍自动化测试工具和优缺点，下面请详细阅读本文吧。

Automation Testing



目录

一、自动化测试方法、流程与工具	1
一、 自动化测试简介	4
1. 为什么我们需要执行自动化测试?	6
2. 可重用性	7
3. 一致性	7
4. 运行测试 24/7	8
5. 早期错误检测	8
6. 降低人员成本	8
二、 自动化测试方法	8
1. GUI (图形用户界面) 测试	9
2. 代码驱动	9
3. 测试自动化框架	10
三、 自动化测试流程	10
1. 步骤 1: 决定自动化测试	10
2. 步骤 2: 测试工具选择	11
3. 步骤 3: 范围介绍	11
4. 步骤 4: 测试计划和开发	12
5. 步骤 5: 测试用例执行	12
6. 步骤 6: 审查和评估	12
四、 在自动化测试过程中面临哪些不同的挑战, 以及如何克服这些挑战?	13
1. 缺乏高技能的测试自动化专家	14

2. 正确选择正确的测试自动化工具.....	14
3. 扩展测试环境的挑战.....	14
4. 包括高昂的初始投资成本.....	15
5. 开发和测试团队之间沟通不畅和缺乏协作.....	15
五、 自动化测试工具.....	15
1. 商业功能测试工具.....	16
2. QTP.....	17
3. RFT.....	18
4. 测试完成.....	19
5. SoapUI.....	20
6. 开源功能测试工具.....	21
7. Selenium.....	21
8. Selenium 的特性.....	22
9. Sikuli.....	23
10. Autolt.....	24
11. 非功能性自动化测试工具.....	25
12. 商用非功能性自动化测试工具.....	25
13. 开源非功能性自动化测试工具.....	28
六、 自动化测试的优势.....	30
七、 自动化测试的缺点.....	31
八、 概述.....	31

在前面关于软件测试的文章中，我们了解到软件测试分为两种类型的测试，分别是手动测试和自动化测试。手动测试和自动化测试都有其特点和方法，使这两种测试技术彼此不同。

在这里，我们将了解自动化测试的以下相关主题：

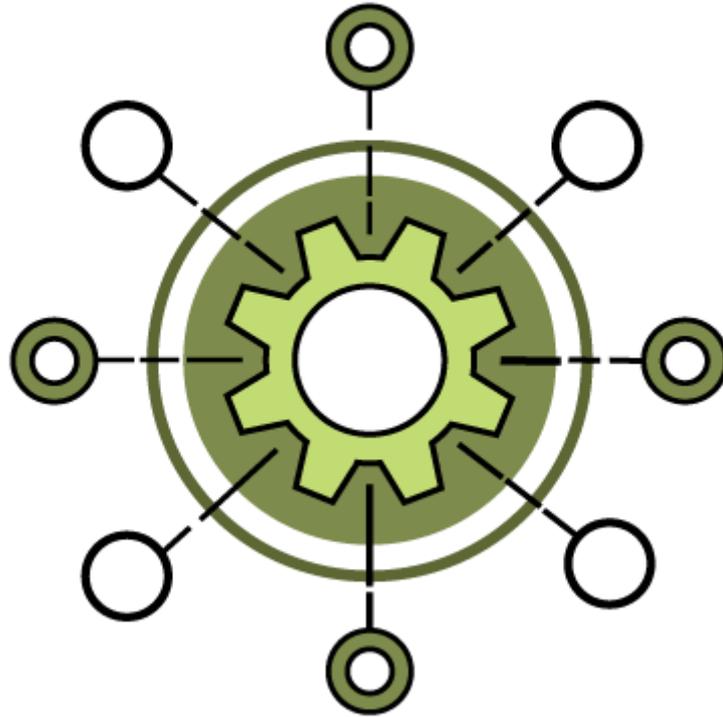
- 自动化测试简介
- 为什么我们需要执行自动化测试？
- 自动化测试中使用的不同方法
- 自动化测试流程
- 在自动化测试过程中面临哪些不同的挑战？
- 自动化测试工具
- 自动化测试的优点和缺点。

一、 自动化测试简介

一种软件测试方法是自动化测试，它使用一些特定的工具来执行测试脚本，而没有任何人为干扰。这是提高软件测试的效率、生产力和测试覆盖率的最可接受的方法。

借助自动化测试工具，我们可以轻松处理测试数据，处理测试实现，并将实际输出与预期结果进行比较。

Automation Testing



在自动化测试中，测试自动化工程师将编写测试脚本或使用自动化测试工具来执行应用程序。另一方面，在手动测试中，测试工程师会编写测试用例，并在书面测试用例的基础上实现软件。

在测试自动化中，测试工程师可以执行重复性任务和其他相关任务。在手动测试中，一次又一次地实现重复拍摄是一个繁琐的过程。

换句话说，我们可以说测试自动化的主要重点是通过系统或设备改变手动人类活动。

自动化测试过程是一个节省时间的过程，因为它在探索性测试上花费的时间更少，而在保留测试脚本上花费的时间更多，同时增强了完整的测试覆盖率。

注意：只要没有要求，我们就会执行探索性测试。

1. 为什么我们需要执行自动化测试?

- 在软件测试中，需要自动化测试来测试应用程序，因为他更便捷的为我们提供了更好的应用程序。
- 一些组织仍然只执行手动测试来测试应用程序，因为这些公司并不完全了解自动化测试过程。
- 但是现在，他们意识到自动化测试并在应用程序开发过程中执行测试自动化过程。
- 为了实施自动化测试，我们需要相当多的资源和金钱投资。

自动化测试的执行为我们提供了各种优势，如下所述：

- 可重用性
- 一致性
- 运行测试 (24/7)
- 早期错误检测
- 减少手动测试人员，降低人员成本



2. 可重用性

我们可以在自动化测试中重用测试脚本，不需要一次又一次地编写新的测试脚本。而且，我们还可以重新创建与前面的步骤一样详细的步骤。

3. 一致性

与手动测试相比，自动化测试比执行常规单调测试更一致，速度更快，后者不能错过，但在手动测试时可能会导致故障。

4. 运行测试 24/7

在自动化测试中，我们可以随时随地开始测试过程。如果我们没有很多方法或购买它们的选择，我们甚至可以远程做到这一点。

5. 早期错误检测

通过执行自动化测试，我们可以轻松检测软件开发过程初始阶段的关键错误。它还有助于我们花更少的工作时间来解决这些问题并降低成本

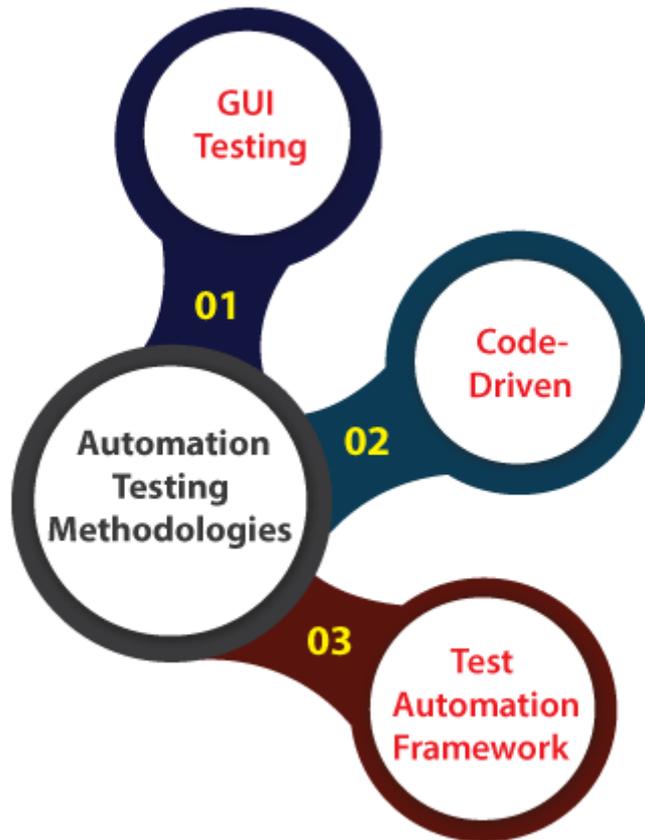
6. 降低人员成本

要实现自动化测试脚本，我们需要一个测试自动化的工程师，他可以编写测试脚本来自动执行测试，而不是让几个人重复执行繁琐的手动测试。

二、 自动化测试方法

自动化测试包含以下三种不同的方法和途径，这将有助于测试工程师提高软件产品质量。

- 图形用户界面测试
- 代码驱动
- 测试自动化框架



现在，让我们一一了解自动化测试的不同方法：

1. GUI (图形用户界面) 测试

在这种方法中，我们可以实现包含 GUI 的软件或应用程序。因此，自动化测试工程师可以记录用户操作并多次评估它们。

我们知道测试用例可以用几种编程语言编写，如 [JAVA](#), [C#](#), [Python](#), [Perl](#) 等。

2. 代码驱动

代码驱动技术是自动化测试中使用的后续方法。在这种方法中，测试工程师将主要集中在测试用例的执行上，以确定代码的几个部分是否按照给定的需求执行。

因此，它是[敏捷软件开发](#)中非常常用的方法。

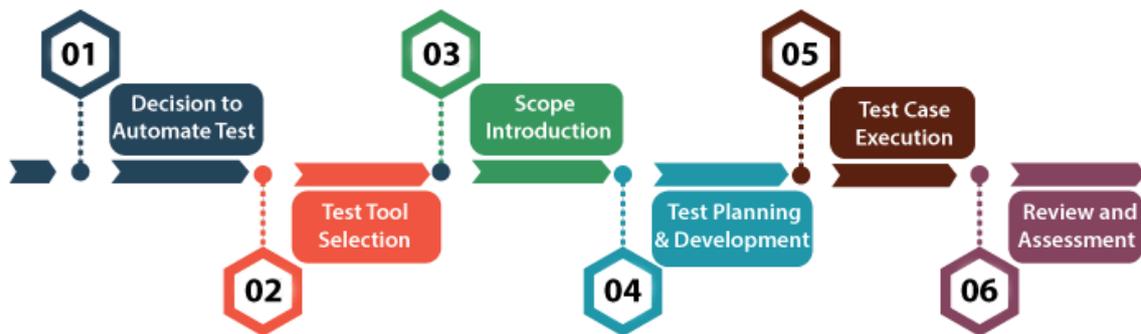
3. 测试自动化框架

自动化测试的另一种方法是测试自动化框架。测试自动化框架是一组规则，用于生成自动化测试活动的有价值结果。

同样，它将测试数据源、函数库、对象详细信息和其他可重用模块结合在一起。

三、 自动化测试流程

自动化测试过程是一种系统的方法，以有限的资源提供最大的测试覆盖率来组织和执行测试活动。测试的结构涉及一个多步骤过程，支持执行任务所需的、详细的和相互关联的活动。



自动化测试过程在以下步骤中完成：

1. 步骤1：决定自动化测试

这是自动化测试生命周期方法（ATLM）的第一阶段。在此阶段，测试团队的主要重点是管理测试的期望，并找出正确应用自动化测试的潜在好处。

在采用自动化测试服时，组织必须面对许多问题，下面列出了一些问题：

- 测试工具专家是需要自动化测试的，所以第一个问题，要指定一名测试设备专家。

- 第二个问题是，选择用于测试特定功能的确切工具。
- 实施自动化测试过程中的设计和开发标准问题。
- 分析各种自动化测试工具，选择最适合自动化测试的工具。
- 由于在测试开始时金钱和时间的消耗很高，因此会出现金钱和时间的问题。

2. 步骤2：测试工具选择

测试工具选择代表自动化测试生命周期方法（ATLM）的第二阶段。此阶段指导测试人员评估和选择测试工具。

由于测试工具几乎支持所有测试需求，因此测试人员仍然需要审查系统工程环境和其他组织需求，然后列出工具的评估参数。测试工程师根据提供的样品标准评估设备。

3. 步骤3：范围介绍

此阶段代表自动化测试生命周期方法（ATLM）的第三阶段。自动化的范围包括应用程序的测试区域。范围的确定基于以下几点：

- 每个软件应用程序所包含的软件应用程序的通用功能。
- 自动化测试设置可重用的业务组件范围。
- 自动化测试决定了业务组件的可重用性程度。
- 应用程序应具有特定于业务的功能，并且必须在技术上可行。
- 自动化测试在跨浏览器测试的情况下提供测试用例的重复。

此阶段可确保整体测试策略，如果需要，应对其进行良好的管理和修改。为了确保技能的可用性，根据特定软件应用程序所需的特定技能分析特定成员和整个团队的测试技能。

4. 步骤4：测试计划和开发

测试计划和开发是自动化测试生命周期方法（ATLM）的第四个也是最重要的阶段，因为所有测试策略都在这里定义。长线索测试活动的计划，标准和指南的创建，硬件，软件和网络组合的安排以创建测试环境，缺陷跟踪程序，控制测试配置和环境的指南都在此阶段确定。测试人员确定整个项目的估计工作量和成本。测试策略和工作量估算文档是此阶段提供的可交付成果。测试用例执行可以在测试计划成功完成后开始。

5. 步骤5：测试用例执行

测试用例执行是自动化测试生命周期方法（ATLM）的第六阶段。它在成功完成测试计划后进行。在此阶段，测试团队定义测试设计和开发。现在，测试用例可以在产品测试下执行。在此阶段，测试团队使用自动化工具启动案例开发和执行活动。准备好的测试用例由测试团队的同行成员或质量保证负责人进行审查。

在测试程序执行过程中，测试团队指示遵守执行计划。执行阶段实现之前在测试计划中定义的集成、验收和单元测试等策略。

6. 步骤6：审查和评估

审查和评估是自动化测试生命周期的第六个也是最后一个阶段，但此阶段的活动贯穿整个生命周期，以保持持续的质量改进。改进过程是通过评估矩阵、审查和评估活动来完成的。

在审查过程中，审查员集中精力确定特定指标是否满足验收标准，如果是，则准备在软件生产中使用。它是全面的，因为测试用例涵盖了应用程序的每个功能。

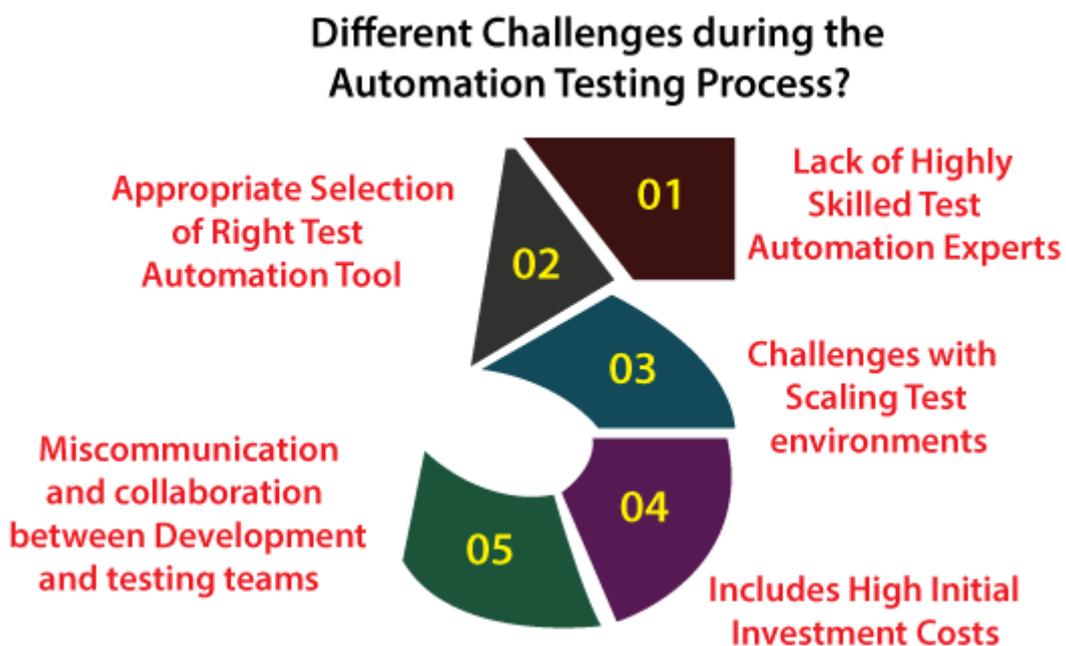
测试团队执行自己的调查，以询问该过程的潜在价值;如果潜在收益不足，测试团队可以更改测试工具。该团队还提供一个示例调查表单，以征求最终用户对软件产品属性和管理的反馈。

四、在自动化测试过程中面临哪些不同的挑战，以及如何克服这些挑战？

自动化测试过程为组织带来了大量优势。通常，通过自动化测试，指定软件验证方法的测试团队可以完成放大的测试覆盖率。

尽管我们在自动化测试过程中可能会面临各种挑战;因此，我们需要一个彻底的后续流程来成功执行测试自动化。

下面讨论的一些常见挑战，我们可能会在整个应用程序的测试自动化过程中遇到这些挑战：



1. 缺乏高技能的测试自动化专家

- 缺乏自动化测试流程的最根本原因是缺乏高技能的自动化测试工程师。
- 为了设计和维护测试自动化框架和测试脚本，测试自动化工程师需要具备技能并具备有关自动化技术技能的详细知识。
- 若要正确创建测试脚本并维护它们，团队应成功实现测试脚本以检查应用程序性能并修复技术问题。
- 为了实现操作自动化方面，自动化测试工程师应该对测试自动化框架有深入的了解，因为他们很容易设计和执行测试脚本。

2. 正确选择正确的测试自动化工具

- 我们在执行自动化测试时主要面临的下一个挑战是选择正确的自动化测试工具。
- 必须对被测应用（AUT）进行适当的详细评估。
- 可以访问开源和付费工具的各种自动化测试工具。在接受工具之前进行适当的检查至关重要。

3. 扩展测试环境的挑战

- 作为不在云中提供测试环境的测试团队，这是一个至关重要的挑战。
- 快速交付自动化测试所需的更改测试环境、测量它们、执行测试以确保在云中测试时取得成功至关重要。
- 通常，内部团队有部分测试环境，他们可以使用这些环境来执行较少的测试，并在任何给定时间部署它们。

- 因此，测试需要更长的时间。为了克服这一点，必须将测试环境迁移到云以扩展测试自动化，并且还将降低测试团队的运营成本。

4. 包括高昂的初始投资成本

- 通常，在初始阶段执行自动化测试过程的成本有点高，因为有必要通过详细的 AUT 分析来评估，设计和构建自动化测试框架，库或可重用功能。
- 此外，如果自动化测试工程师选择许可的测试自动化工具，则应评估运营成本。
- 如果自动化测试工程师选择开源工具，那么也应该花费大量精力在学习、训练和维护上。
- 之后，自动化工具的选择，无论是开源的还是许可的，都取决于 AUT 优先级和成本设施的可用性。

5. 开发和测试团队之间沟通不畅和缺乏协作

- 开发人员与测试工程师的团队和运营团队之间的沟通不畅或缺乏适当的协作并不能使我们成功实现测试自动化。
- 因此，将所有团队（开发人员、测试、操作）成员安排起来对测试自动化目标进行分类并设定目标非常重要。
- 为了确保测试自动化的成功，这些团队需要花一些时间来进行有效的沟通，并清楚地了解业务需求和项目规范。

五、 自动化测试工具

自动化测试工具可以分为两类，如下所示：

- 功能测试工具
- 非功能性测试工具

Automation Testing Tools



让我们——详细了解：

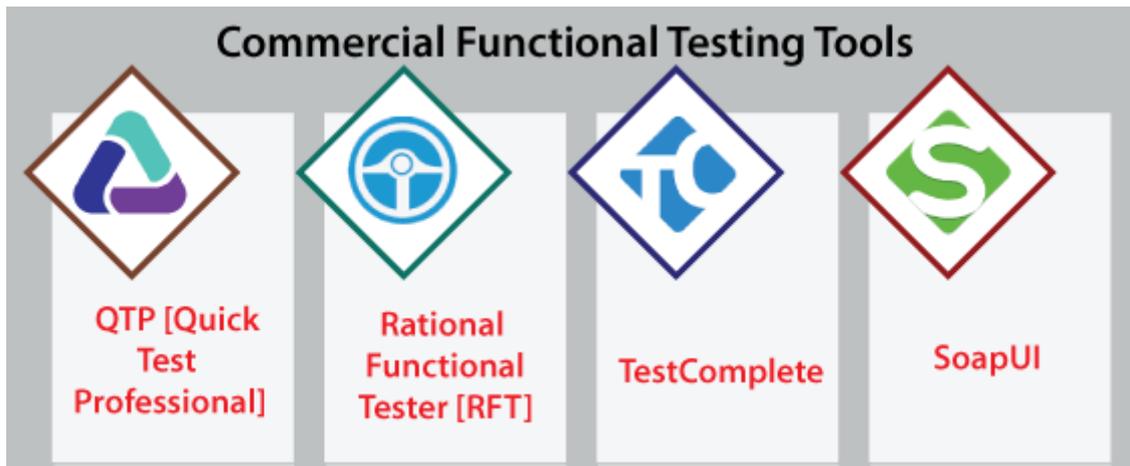
1. 商业功能测试工具

商业**功能**测试工具是那些在市场上不能免费提供的测试工具。这些工具也称为**许可工具**。

与开源工具相比，许可工具包括各种功能和方法。

一些最重要的商业工具如下：

- QTP[快速测试专业版]
- 理性功能测试仪 [RFT]
- 测试完成
- 肥皂用户界面



让我们来看看上面列出的商业工具：

2. QTP

- QTP 表示快速测试专业，但现在它被称为 Micro FocusUT（统一功能测试）。
- 它主要用于自动化基于 Web 的应用程序或软件的功能回归测试用例。
- 为了测试应用程序、部署对象并出于分析目的，它是在脚本语言（如 VBScript）上设计的。
- 它遵循关键字驱动测试的概念来定义测试的创建和维护。
- 新的测试工程师可以轻松使用 QTP 工具，因为它允许他们直接从应用程序开发测试用例。



QTP 的特点

QTP 的一些最标准属性如下：

- 录制和回放功能增强了 QTP 工具的可信度。
- 技术和非技术测试工程师都可以操作 QTP 工具。
- 它支持多种软件开发环境，如 SAP, JAVA, Oracle 等。
- 我们可以使用 QTP 检查基于 Web 和基于桌面的应用程序。

3. RFT

- RFT 代表 Rational Functional Tester，用于执行功能回归测试用例。
- 它遵循数据驱动测试和 GUI 测试的概念。
- 它通过记录最终用户在被测系统上的活动并根据请求重复操作以执行测试来构建自动化功能测试。



RFT 的特点

RFT 工具包含以下特征：

- RFT 支持广泛的应用程序和协议，如 HTML, Java, .NET, Windows, Eclipse, SAP, Visual Basic, PowerBuilder
- 它通过代理 SDK（如 Java or.Net）维护自定义控件。

- 它允许开发人员开发与关键字相关的脚本;因此,它可以再次重复使用,从而提高项目的生产力。

4. 测试完成

- 另一个功能性自动化测试工具是 TestComplete, 它被 SmartBear Software 收购。
- TestComplete 具有一些内置功能, 可帮助测试工程师为 Microsoft Windows、Web、Android 和 iOS 应用程序开发自动化测试。
- 在此工具中, 可以使用关键字驱动的进程记录、编写脚本或手动创建测试, 并用于自动回放和错误记录。
- 它包括三个不同的模块, 如 Web, 桌面和移动, 其中所有模块都涉及在特定平台上开发自动化测试的功能。



测试完成工具的特点

以下是测试完成工具的基本特征:

- 它用于测试几种类型的特定应用程序, 如 Web, Windows, Android, iOS, .NET, VCL 和 Java。
- 它涉及问题跟踪模板, 可用于开发或更改存储在问题跟踪系统中的项目。
- 它记录了重复测试的关键操作, 并拒绝所有不必要的努力。

5. SoapUI

- SoapUI 是使用最广泛的自动化工具。它主要用于测试 SOAP 和 REST 接口的 Web 服务和 WebAPI。
- 它允许测试工程师在各种 Web 服务和 API 上测试功能、回归测试和其他测试类型。
- SoapUI 工具开发了模拟，测试工程师可以在其中测试实际应用程序。
- 它完全是用 JAVA 和 Groovy 编程语言编写的。



SoapUI 的特点

SoapUI 工具的一些最常见的特征如下：

- SoapUI 的核心功能包含各种 Web 服务，例如开发、模拟、检查、模拟和调用。
- 它支持所有标准协议和技术，如 HTTP, HTTPS, AMF, JDBC, SOAP, WSDL 等。
- SoapUI 工具提供了一个快速高效的框架，可以创建许多 Web 服务测试。
- 它提供了从多个 Web 服务源获取数据的服务，而无需开发任何代码。

要获取有关 SoapUI 工具的更多信息，请参阅以下链接：

<https://www.javatpoint.com/soapui>。

6. 开源功能测试工具

开源功能测试工具是那些在市场上免费提供的工具。这些工具的功能和特性比商业/许可工具少，但有时在商业工具上工作成为一个昂贵的过程。

这就是为什么一些知名组织更喜欢使用开源工具的原因。

以下是最常用的开源功能自动化测试工具：

- Selenium
- Sikuli
- Autoit

我们来——了解一下开源功能自动化测试工具：



7. Selenium

每当我们谈论自动化测试的开源工具时，每个自动化测试工程师都会想到一个名字：

Selenium。

- Selenium 是强烈推荐和广泛使用的功能测试工具，非常适合非功能测试工具。
- Selenium 是一个开源工具，这意味着它不需要任何许可。
- 我们只能使用 Selenium 工具测试 Web 应用程序，而独立应用程序无法在 Selenium 中自动化。
- 它最常用于实现功能测试脚本。
- 它可以与几种自动化测试工具结合使用，以实现连续测试，例如 Maven 和 Jenkins。
- 它可以与 TestNG 和 JUnit 等许多设备相关联，以管理测试用例并生成测试报告。



8. Selenium的特性

以下是 Selenium 的基本特征：

- Selenium 可以很容易地部署在 Windows, Linux, Solaris 和 MacOS 等多个平台上。
- 此外，它还支持移动应用程序的操作系统，例如 iOS, Windows 移动和 Android。
- Selenium 支持多种编程语言，如 C#, Java, Perl, PHP, Python 和 Ruby。
- 当我们实现测试脚本并同时跨多个浏览器分析它时，Selenium 变得非常足智多谋。

- 它提供了一个用户友好的界面，可帮助测试工程师高效地开发和执行测试脚本。

要获得有关 Selenium 的完整信息，请参阅以下链接：

<https://www.javatpoint.com/selenium-tutorial>。

9. Sikuli

- 另一个开源的功能自动化测试工具是 Sikuli。
- 它是一个基于 GUI 的测试自动化工具，可以轻松地自动化 Flash 对象，因为大多数自动化测试工具（如 Selenium）不支持 Flash 对象自动化。
- 最常见的是，Sikuli 用于与网页的基本原理进行交互并控制基于窗口的弹出窗口。
- 借助 Sikuli 工具，我们可以轻松测试 Windows 应用程序。



Sikuli 的特点

Sikuli 工具的最标准功能如下所述：

- Sikuli 工具可以很容易地与 Selenium WebDriver 和所有其他自动化测试工具结合使用，通过使用 Sikuli Jar 文件。
- 我们还可以在 Sikuli 工具的帮助下自动化桌面应用程序或独立应用程序。
- 它提供了一个简单的 API，这意味着可以使用屏幕类对象检索所有方法。

注意：众所周知，Selenium WebDriver 只能测试 Web 应用程序，但另一方面，Sikuli 工具可以测试基于 Web 和基于窗口的应用程序。

10. Autolt

- Autolt 是另一个用于功能自动化测试的开源工具。
- 它是一种免费软件脚本语言，旨在测试 Windows GUI 和通用脚本。
- Autolt 脚本是用主要语言编写的。
- 它可以复制击键、鼠标移动以及窗口或控件操作的任何分组。



汽车 IT 的特点

AutoIT 工具包括以下功能：

- 它的脚本语言使用简单的语法，可以轻松理解和执行以自动化任何过程。
- 它有一个内置的 RunAs 函数，有助于使用一些外部用户执行任何外部程序。
- 它甚至可以自行记录和创建脚本，用于需要自动化的详细流程。
- 没有任何对象识别问题的所有类型的标准 Windows 控件和其他 GUI 都可以轻松与 Autolt 工具相关。

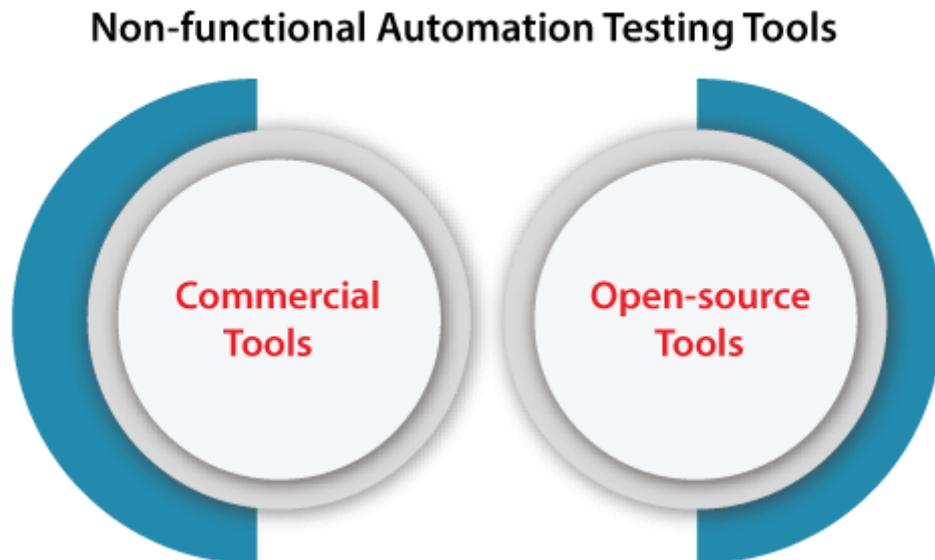
11. 非功能性自动化测试工具

自动化测试工程师使用非功能性自动化测试工具来执行非功能性性能测试用例。

例如，在重大负载（例如 100 个用户）下测试应用程序的响应时间。

就像功能自动化测试工具一样，非功能自动化测试工具分为两个不同的类别，具体如下：

1. 商业工具
2. 开源工具



12. 商用非功能性自动化测试工具

这些是那些不能自由使用的工具，因为它们需要适当的许可证。与其他开源测试工具相比，商业工具具有额外的特性和功能。

使用这些类型的工具有助于我们提高软件产品的效率。

让我们看看一些最常用的商业非功能性自动化测试工具。

- **负载运行器**
- **丝绸表演者**

负载运行器 [HP 性能测试仪]

LoadRunner 是最流行的**非功能性**工具之一。它主要用于支持各种协议、多种技术和应用程序环境的性能测试。LoadRunner 是许可工具。

它可以快速对最大数量的性能问题进行分类。并准确预测应用程序的可扩展性和大小。



负载运行器的特点

LoadRunner 工具的一些最重要的功能如下：

- LoadRunner 工具将帮助我们降低分布式负载测试的成本以及硬件和软件成本。
- 我们可以轻松地查看和处理测试脚本中的 XML 数据，因为它支持 XML 脚本语言。
- 我们可以通过使用 LoadRunner 工具获得详细的性能测试报告。
- 它提供了用于部署跟踪的操作工具。

Silk Performer 另一个非功能性自动化测试工具是 Silk Performer。它可以测试具有数千个并发用户的各种应用程序环境。

它确保在面对最高的客户使用率时，应用程序和服务器的正常运行时间得以维持。

它是最常用的企业级负载和压力测试工具之一，支持广泛的协议。



Silk Performer 的特点

以下是 Silk 执行器工具的标准功能：

- 它用于假装可变的虚拟用户。
- 在此工具中，关联和参数化是用户友好的。
- 它支持集成的服务器监控。
- 它提供版本控制管理。
- 控制器或单个协议不需要许可条件。
- 它可以快速创建带有表格和图形的报告，并允许自定义。

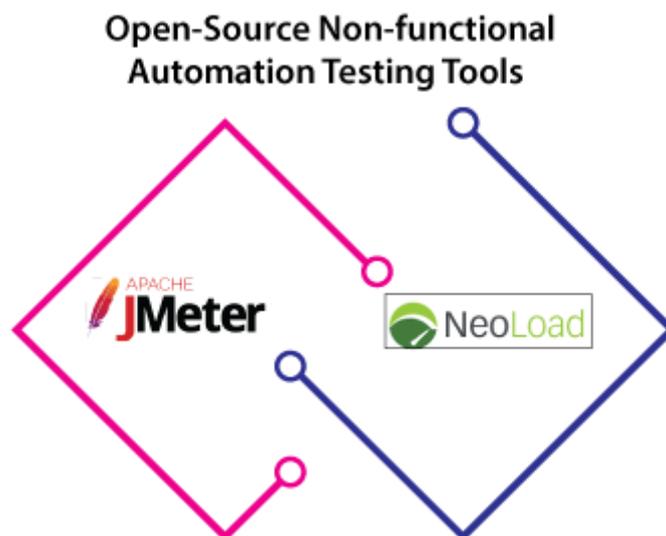
13. 开源非功能性自动化测试工具

开源非功能性自动化测试工具可以轻松使用，因为它们在市场上免费提供。与商业测试工具相比，这些工具的功能较少。

但是由于它们是开源的并且使用速度很快，因此一些组织更喜欢使用这些工具，因为它们不需要任何成本。

一些最常用的开源非功能性自动化测试工具如下：

- JMeter
- 新加载



JMeter

JMeter 是使用最多的开源非功能性自动化测试工具之一。JMeter 完全在 JAVA 应用程序上设计，以加载高效的测试性能并衡量应用程序的生产力。

它主要是简化用户或开发人员使用源代码来增强其他应用程序。它用于测试静态和动态资源以及动态 Web 应用程序的实现。

它用于重现服务器、对象或网络、服务器组上的大量负载，以测试其功率或探索多种负载类型的完整性能。

在几种类型的服务器中加载性能测试具有高度可扩展性，例如：

- 数据库服务器：LDAP、JDBC
- Web Server： SOAP, HTTPS, HTTP
- 邮件服务器：POP3



JMeter 特点

JMeter 工具的一些重要属性如下：

- 它支持多种测试方法，例如功能测试、分布式测试和负载测试。
- 它支持交互式且简单的用户友好型 GUI。
- 它是一个独立于平台的工具，因为它是在 JAVA 的帮助下开发的;因此，它可以在任何接受 JVM 的平台上执行，如 Windows, Mac 和 Linux 等。

有关 JMeter 的更多信息，请参阅以下链接：

<https://www.javatpoint.com/jmeter-tutorial>。

NeoLoad

自动化测试中另一个最常用的开源工具是 NeoLoad，Neotys 开发了它。

它用于测试性能测试场景，还帮助我们识别 Web 和移动应用程序开发过程中的瓶颈区域。

与其他传统工具相比，它更快。NeoLoad 将支持广泛的 Web、移动和打包应用程序，如 SAP、Oracle、Salesforce 等，涵盖我们所有的测试需求。



NeoLoad 的特点

以下是 NeoLoad 的一些重要功能：

- 它还用于共享和管理测试资源。
- 它将支持各种框架和协议，如 HTTP / 2, HTML5, API, AngularJS, Web Socket, SOAP 等。
- 它提供了强大的无代码设计。
- NeoLoad 可以将功能测试脚本修改为性能测试脚本。
- NeoLoad 可以自动更新测试脚本并开发实时测试结果。

六、 自动化测试的优势

- 自动化测试比手动测试花费更少的时间。
- 如果重复执行多次相同的操作，测试人员可以测试软件的响应。
- 自动化测试提供了测试用例对同一软件不同版本的测试的可重用性。
- 自动化测试是可靠的，因为它通过以相同的方式再次执行测试用例来消除隐藏的错误。
- 自动化测试是全面的，因为测试用例涵盖了应用程序的每个功能。
- 它不需要很多人力资源，而不是编写测试用例并手动测试它们，他们需要一个自动化测试工程师来运行它们。
- 自动化测试的成本低于手动测试，因为它需要一些人力资源。

七、 自动化测试的缺点

- 自动化测试需要高水平的熟练测试人员。
- 它需要高质量的测试工具。
- 当遇到不成功的测试用例时，对整个事件的分析是复杂的。
- 测试维护费用昂贵，因为需要高费用的许可证测试设备。
- 如果尚未解决效果较差的错误，则必须进行调试，这可能会导致致命的结果。

八、 概述

在本教程中，我们了解了自动化测试，自动化测试方法，自动化测试过程，自动化测试工具，自动化测试过程中的不同挑战，自动化测试的优缺点。

最后，我们可以得出结论，自动化测试是一种软件测试技术，它是在特殊的自动化测试软件工具的帮助下实现的。

这是执行测试用例套件的最佳方法，这有助于我们提高软件测试的测试覆盖率、效率和性能速度。

自动化测试工具、测试流程和团队的选择是成功实现自动化的关键方面。

自动化测试高度依赖于构建待测应用程序所基于的技术。

测试自动化维护方法是一个自动化测试阶段，用于测试添加到软件中的新功能是否正常工作。

为了成功测试软件，手动和自动化技术是齐头并进的。我们应该清楚自动化过程，它用于减少某些类型测试的测试时间。