

作者



FunTester



FunTester

FunTester，Have Fun ~ Tester ！

61

3

文章

0

精华

27

0

粉丝

FunTester 自动化的 10 项准备工作

FunTester · 2024年01月17日 · 2458 次阅读

「原创声明：保留所有权利，禁止转载」

软件或应用程序的测试工作和开发人员创建软件的工作同样重要。有时候，软件测试可能被忽视，但它对于公司来说是至关重要的，可以避免不必要的尴尬和金钱损失。软件测试通常需要进行重复的测试，因为任何人为的错误都可能影响测试结果。为了避免这样的错误，如今大多数公司选择将测试自动化与 DevOps 相结合。这意味着每次向现有代码库添加新代码时，测试人员不再需要重复相同的步骤和操作，也不需要重新从头开始测试软件。通过使用诸如 Selenium 这样的软件，自动化测试变得越来越受欢迎。

目录

在实施自动化测试之前，以下是需要了解的几项重要原则。

自动化需要大量投入

虽然测试自动化可以节省时间，但它是一项需要规划和专注的工作。这需要全职投入。测试人员不能同时进行手动测试和自动化测试。

首先，需要确定哪些测试用例可以被自动化。这包括识别业务关键、重复和单调的测试用例。然后需要确定应用程序的哪些部分可以被测试。接下来是规划阶段——选择适合的测试框架、制定时间表，明确测试可能揭示的结果等。

是否可以自动化

并非每个测试都需要进行自动化。有些测试可能根本无法自动化，或者自动化测试无法捕捉到所有类型的错误。因此，有必要自动化那些您确信能够减少工作量的测试。

回归测试是可以自动化的。每次向代码库中引入新代码时，需要对整个应用程序进行测试，以确保新代码不会破坏现有功能。这种回归测试可以通过自动化来完成。同样，对于需要输入大量数据的测试来说，很容易出现人为错误，因此这些测试应该被自动化。如果您需要了解应用程序在不同操作系统和浏览器中的表现，测试自动化可以帮助您模拟这种情况。

有时候，需要测试网络应用程序，确保其在繁忙时期能够承受数千用户的请求和操作。但是，如果您只是一个独立的测试人员，或者您的测试团队只有 7–8 人，您如何测试 Web 应用程序的负载能力呢？这就是自动化发挥作用的地方。自动化测试可以创建数千个虚拟用户，模拟数千人同时使用网络应用程序的情景。

然而，并非所有测试都适合自动化。有些应用程序方面需要手动测试人员来检查是否存在问题。一个典型的例子就是用户界面。只有人类测试人员才能辨别 UI 中可能存在的视觉缺陷。例如，假设在登录界面引入了背景图像，但图像上的标语存在拼写错误。这种错误是无法通过自动化测试发现的，因为该错误不是在网站上的文字部分，而是在图像

上。只有手动测试人员才能发现这种错误。

消除结果中的歧义

在软件开发中，自动化测试是确保代码质量和功能稳定性的关键环节。在进行自动化测试时，测试结果的准确性和一致性至关重要。有时候，测试结果可能出现不明确或不一致的情况，这可能是测试脚本或计划存在问题的迹象。

自动化测试旨在验证软件的各个方面，确保它们按照预期工作。然而，当测试结果不明确时，可能会导致测试过程的不确定性。例如，如果测试在一次执行中通过，但在另一次执行中失败，这种不一致性可能暗示着测试环境、测试数据或测试脚本本身存在问题。

一个成熟的自动化测试流程应该是可靠的，它的结果应该是稳定和可预测的。如果测试结果在多次运行中不一致，这可能会影响对软件质量的评估和信任度。这样的结果会给团队带来困扰，并且可能导致额外的工作量，以识别和解决问题。

对于自动化测试而言，保持一致的结果是至关重要的。任何不一致性都可能是测试流程或脚本中潜在问题的指示。这可能包括但不限于：测试环境的变化、依赖项的状态、数据准备问题或测试脚本的逻辑错误。发现并解决这些问题是确保测试的稳定性和可靠性的关键步骤。另一方面，即使自动化测试结果是稳定的，也不能完全排除软件本身存在问题的可能性。因此，测试人员需要综合考虑多种因素，包括测试结果、软件功能和代码质量，以确保最终交付的软件是高质量、稳定可靠的。

自动化测试的可靠性和一致性对于确保软件质量至关重要。不明确或不一致的测试结果应该被视为潜在问题的信号，并需要进一步的审查和调查，以确保测试流程的有效性和稳定性。

优先设计用例

经过深思熟虑并精心设计的测试用例是测试自动化场景中成功检测错误的关键。如果没有适当的规划，您的测试用例可能会失去效用。

测试设计过程有多种方法。其中一种流行的方法是黑盒测试技术。在这种技术中，边界值分析 (BVA) 是一个常用的方法。它涉及测试输入值的最边缘情况，以验证这些值如何与应用程序互动。边界处往往容易出现错误。例如，假设 Facebook 希望排除 18 岁以下的用户，您可以使用这个测试用例来验证这个排除过程是否有效。这个测试用例将包括 6 个用例，分别检查值是否为 18、19、129、130、17 和 131（假设 Facebook 接受的最大年龄为 130 岁）。

另一个方法是等价类划分 (EP)。如果有大量值需要测试以确定它们的有效性，EP 可以将这些值分类为不同的类别，从而减少测试工作量。

还有一种测试设计类型是决策表测试。这种测试首先检查是否满足所需输出的所有条件。此外，还有其他黑盒测试技术，例如状态转换图或用例测试，这些以后有机会在讨论。

降低使用门槛

使用测试团队熟悉的编程语言进行测试自动化是非常重要的原则。如果应用程序是用 Python 编写的，并且您的团队成员精通 Python，那么应该选择 Python 进行测试自动化以确保一致性。然而，如果应用程序是用测试团队不太了解的编程语言编写的，您可以评估测试团队是否可以轻松学习并掌握这种语言。如果团队能够快速学习并感到自如，那么可以考虑使用该语言进行测试自动化。但如果测试人员对该语言不熟悉或学习成本很高，那么应该选择测试人员熟悉的语言来进行测试自动化。重要的是确保团队对

所选择的语言感到舒适和自信，以确保测试自动化工作的高效性和准确性。

了解应用程序

在制定测试自动化计划之前，充分了解待测试的应用程序是至关重要的。这个过程需要收集详尽的信息，以确保选择合适的工具和方法来执行自动化测试。无论是 Web 应用、桌面应用还是移动应用，每种应用类型都有其独特的特征和测试需求。

首先，需要明确目标应用程序的类型。例如，如果是一个 Web 应用程序，它可能是基于浏览器的，这时候就可以考虑使用像 Selenium 这样的自动化测试工具。Selenium 是一个强大的工具，专门用于测试 Web 应用程序的功能和用户界面。然而，如果应用程序是桌面应用，Selenium 并不适用于这种场景。对于桌面应用程序，可能需要考虑其他专门针对桌面应用的自动化测试工具或方法。

另一个重要的考虑因素是应用程序支持的平台。不同的应用程序可能在多个操作系统（如 Windows、macOS）或多个移动设备平台（如 iOS、Android）上运行。因此，在选择自动化测试工具时，需要确保其支持目标应用程序所运行的平台。对于跨平台应用程序，可能需要考虑具备跨平台支持的测试工具。

除了应用程序类型和支持的平台外，还需要了解应用程序的特性和功能。不同的应用程序可能具有不同的功能模块、用户交互方式、数据输入和输出等方面的特点。这些特性将直接影响测试策略和自动化测试的实施。因此，在测试自动化计划中，必须考虑到这些方面，并选择适合特定功能和特性的测试工具和方法。还需要考虑可能需要的测试工具和技术。这可能包括测试框架、集成开发环境（IDE）、模拟器、虚拟机等。根据应用程序类型和特性，选择合适的工具和技术来支持自动化测试流程。

在制定测试自动化计划之前，收集关于应用程序的所有相关信息是非常关键的。这种全面的了解能够帮助您选择最合适的测试工具和方法，以确保测试流程高效、全面并且能够覆盖应用程序的各个方面。

提升健壮性

在测试自动化过程中，测试用例之间的相互关联是至关重要的考虑因素。测试用例的依赖性意味着如果前面的测试用例失败，可能会对后续测试用例的执行造成影响。因此，将测试套件划分为可以独立执行的小块是一个明智的做法，这有助于确保每个测试用例或测试集都能够独立地运行。

通过将测试用例分解为相对较小和独立的单元，能够更好地管理和控制测试用例之间的关系。这种分割方法有助于隔离问题，降低一个测试用例失败对其他用例造成的影响。当测试用例能够独立执行时，即使某个测试失败，也不会对整个测试套件产生灾难性的影响。

此外，小型独立的测试单元使得问题定位和修复更加容易。当测试用例被分割成小块时，可以更精准地确定出问题所在，因为问题通常局限于特定的单元。这有助于加快问题解决速度，提高测试用例的可靠性和稳定性。

总的来说，将测试用例划分为小而独立的单元有助于识别和隔离问题，减少失败测试用例对其他测试用例的影响。这种方法有助于提高测试流程的鲁棒性，同时使得问题的定位和解决更加高效。

高质量测试数据

构建高质量的测试数据是确保软件测试有效性和全面性的关键要素。测试数据在软件测试过程中扮演着至关重要的角色，它们决定了测试覆盖范围的广度和深度，直接影响着测试的质量和准确性。

高质量的测试数据应该是全面的。这意味着测试数据应该覆盖各种可能的情况和使用场景，包括边界情况、异常情况以及各种输入组合。通过考虑不同的情况和变化，测试数据可以更全面地评估软件的稳定性和鲁棒性。测试数据应该是真实可靠的。使用真实世界的的数据能更好地模拟实际使用情况，帮助发现潜在的问题和漏洞。确保测试数据的真实性和可靠性有助于准确评估软件的性能和可靠性。

测试数据应该是多样化和可复用的。多样化的测试数据能够覆盖更多的情况和场景，确保测试的全面性。同时，可复用的测试数据能够在不同阶段和不同测试场景中被重复使用，提高测试效率和成本效益。此外，高质量的测试数据应该是可管理和可维护的。有效管理测试数据意味着对测试数据进行组织、分类和版本控制，以便快速且准确地访问所需的数据。另外，定期维护测试数据，更新和优化数据集，以反映软件的最新变化和需求。

在构建高质量的测试数据时，需要注意数据的准确性和完整性。精心设计和准备的测试数据能够提供更有力的测试覆盖，发现潜在的缺陷和问题。同时，对测试数据的质量进行持续的监控和改进，有助于提高测试的可靠性和有效性。构建高质量的测试数据对于软件测试至关重要。优质的测试数据能够确保测试的全面性、准确性和可靠性，帮助发现和解决软件中的潜在问题，为软件的质量和稳定性提供有力保障。因此，在测试策略中，重视并投入足够的资源和精力来构建和维护高质量的测试数据是至关重要的一环。

不能排除探索性测试

自动化测试通常遵循预设的测试脚本边界，不太会超出这些设定。这种严格的限制有时可能限制了测试的灵活性。相比之下，手动测试人员会结合应用程序的历史弱点，凭借自己的逻辑和直觉进行测试。

比如，假设一家社交媒体公司推出了一个新功能，可以通过在搜索框中输入朋友的电话号码来找到现实生活中的朋友。手动测试人员可能只需关注手机号码是否需要输入国家/地区代码才能得到正确的结果。之后，我们可以将手动测试人员完成的逻辑测试内容纳入测试用例中，看看是否能将其自动化执行。

构建预警系统

在自动化测试中，建立高效的预警系统是确保测试流程有效性和稳定性的重要步骤。这样的系统能够及时发现和处理潜在的问题，提高测试的效率和可靠性。

建立预警系统的总体目标是提早识别测试过程中可能出现的问题或异常情况。这包括监控测试执行过程中的关键指标，如执行时间、错误率、资源利用率等。通过设置阈值和规则，一旦超出预设的范围，系统将自动触发警报，提示测试团队可能存在的风险或问题。

高效的预警系统应该具备及时响应和快速处理问题的能力。它能够迅速通知相关人员，并提供详细的信息以帮助定位问题。及时的警报能够让团队尽早介入、解决问题，避免问题扩大化，确保测试流程的顺利进行。在建立预警系统时需要考虑持续改进和优化。这包括对系统的监控规则、阈值设置和警报机制进行定期检查和调整。随着测试环境和需求的变化，预警系统也需要不断优化，以适应新的挑战 and 情境。

建立高效的预警系统是自动化测试过程中至关重要的一环。它能够帮助团队及时发现并解决问题，提高测试流程的质量和效率。因此，在自动化测试中，投入资源和精力来建立和维护一个高效的预警系统是至关重要的。

- “
- [2021 年原创合集](#)

▪ [2022 年原创合集](#)

▪ [2023 年原创合集](#)

▪ [接口功能测试专题](#)

- [性能测试专题](#)
- [Groovy 专题](#)
- [Java、Groovy、Go、Python](#)
- [单测&白盒](#)
- [FunTester 社群风采](#)
- [测试理论鸡汤](#)
- [FunTester 视频专题](#)
- [案例分享：方案、BUG、爬虫](#)
- [UI 自动化专题](#)
- [测试工具专题](#)

TesterHome 为用户提供「保留所有权利，禁止转载」的选项。 除非获得原作者的单独授权，任何第三方不得转载标注了「原创声明：保留所有权利，禁止转载」的内容，否则均视为侵权。 具体请参见[TesterHome 知识产权保护协议](#)。



如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

打赏支持

暂无回复。

需要 [登录](#) 后方可回复, 如果你还没有账号请点击[这里](#) [注册](#)。



[关于](#) [活跃用户](#) [中国移动互联网测试技术大会](#) [反馈](#) [Github](#) [API](#) [帮助推广](#)
TesterHome社区，测试之家，由众多测试工程师组织和维护的技术社区，致力于帮助新人成长，提高测试地位，推进质量发展。Inspired by RubyChina
[友情链接](#) [WeTest](#)[腾讯质量开放平台](#) / [InfoQ](#) / [掘金](#) / [SegmentFault](#) / [测试窝](#) / [百度测试吧](#) / [IT 大咖说](#)
[简体中文](#) / [正體中文](#) / [English](#)

