

常用软件测试技术介绍

1、灰度测试

灰度测试，是在某项产品或应用正式发布前，选择特定人群试用，然后逐步扩大其试用者数量，以便及时发现和纠正其中的问题。另一个是为了降低同时大量的访问对服务器资源造成压力。常见的灰度测试场景如 APP 新版本升级，系统升级等。

灰度测试主要是按照一定策略选取部分用户，让他们先行体验新版本的应用进行验证。该方法具有用户流量随机，在真实的用户和场景下得到验证，可以实时监控崩溃、用户反馈等问题进行召回、进行定位，进而决定继续放大新版本投放范围直至全量升级或回滚至老版本。灰度测试覆盖达到了机型、系统、网络、地域多类型的目的。

2、A / B 测试

A / B 测试的概念来源于生物医学的双盲测试，双盲测试中病人被随机的分成两组，在不知情的情况下分别给予安慰剂和测试用药，经过一段时间的实验后再来比较这两组病人的表现是否具有显著的差异，从而决定测试用药是否有效。

A / B 测试制定网页或应用程序的两个或多个版本，是在同一时间维度，让一部分用户使用 A 方案，同时另一部分用户使用 B 方案，然后收集统计并对比不同方案的转化率、点击量、留存率等指标，以判断不同方案的优劣并进行决策，分析评估最终哪个版本被采用的方法。

3、模糊测试

模糊测试 (Fuzz testing) 是一种软件测试技术，其核心思想是自动或半自动的生成随机数据输入到一个程序中，并监视程序异常，如崩溃，断言失败，以发现可能的程序错误，比如内存泄漏。模糊测试常常用于检测软件或计算机系统的安全漏洞。

4、探索性测试

探索性测试是敏捷开发过程中的一种重要测试方法,是功能测试和自动化测试的重要补充。探索性测试是一种经过深思熟虑的测试方式,没有测试脚本,可以使你的测试超出各种明显已经测试过的场景。

相对于传统软件测试过程中严格的“先设计,后执行”来说,具有很大区别。另外探索性测试不是自由测试,而是需要有一定的方法来指导。探索性强调测试人员的主观能动性,抛弃繁杂的测试计划和测试用例设计过程,强调在碰到问题时及时改变测试策略。

探索性测试可以用来找到系统深层次的缺陷、也可以加深测试人员对被测系统的了解。但是特别要注意不要将探索性测试和错误推测法、随机测试、回归测试混淆,同时探索性测试不能用来评估软件质量。

5、基于模型的测试用例设计

基于模型的测试 (Model-Based-Testing) 是一种技术,是自动化测试的一个分支,有时被标榜为“自动化测试设计”。它是将测试用例的设计依托于被测系统的模型,并基于该模型自动生成测试用例的技术。

没有测试模型,基于模型的测试工具就无法生成任何测试用例。没有好的测试模型,该工具就无法生成好的测试用例。因此,基于模型的测试技术,是从测试设计变成了测试模型设计。