43 | 发挥人的潜能：探索式测试

2018-10-05 茹炳晟

软件测试52讲 进入课程 >

讲述：茹炳晟
时长 13:16　大小 6.08M

你好，我是茹炳晟。今天我和你分享的主题是：发挥人的潜能之探索式测试。

从今天开始，我们又要一起进入一个新的系列了：测试新技术系列。在这个系列里，我将通过 5 篇文章，和你分享软件测试领域中比较新的 5 个测试主题：探索性测试、测试驱动开发（TDD）、精准测试、渗透测试，以及基于模型的测试。这五种新的测试技术，是我精挑细选的，初衷就是希望帮你拓宽知识面、思路。

今天这次的分享，我们就先从当下很热门的探索式测试开始吧。此时，你可能已经听说过了探索式测试，也很可能还不知道什么是探索式测试。这一切都没有关系，相信你经过我今天的这次分析，总能汲取到新的知识，对探索式测试有一个全面、清晰的认识。

软件测试与招聘面试类比

在正式开始介绍探索式测试之前，我们先一起看一个工作中招聘面试的例子吧。

假设，你是面试官，现在有一个候选人要应聘你负责的这个职位。那么，你通常都会在正式面试前，先仔细了解候选人的简历，然后根据简历情况以及这个职位的要求，设计一些高质量的面试问题。当然了，你这么做的目的是，试图通过这些面试问题判断候选人与这个职位的匹配程度。

但是，在实际面试的时候，你提出面试问题之后，通常会根据面试者的回答调整接下来的问题：

> 如果候选人的回答符合你的预期，你可能就会结束这个话题，然后开始一个新的话题去考察候选人其他方面的能力；
>
> 如果你对候选人的回答有疑问，你很可能就会顺着候选人的回答进行有针对性的提问，这时你提出的新问题完全可能是即兴发挥的，而不是事先准备好的。

事实上，你在面试候选人的过程中，其实同时在开展候选人评估、面试问题设计、面试执行和面试问题回答评估。可以说，这个面试过程其实就是你在“探索”你的候选人，通过“探索”去判断候选人是否符合你的要求。

那么对于软件测试来说，也有非常类似的过程。

比如，你首先根据软件功能描述来设计最初的测试用例，然后执行测试；测试执行后，可能你得到的输出和预期输出不完全一致，于是你会猜测这种不一致是否可能是软件的缺陷造成的；为了验证你的想法，你会根据错误输出设计新的测试用例，然后采用不同的输入再次检查软件的输出。

在一次测试中，你可能会经过几轮这样的猜测和验证，进行反复“探索”，最终确定了一个软件的缺陷。而这个过程中，你会发现，识别缺陷的思路和测试用例的设计，并没有出现在最初的测试设计和测试用例文档中，而是以很快的速度在你的脑海中以及实际测试执行和验证中快速迭代。

从本质上来看，上述的两个过程就是探索式测试最基本的思维模型了。所以说，探索式测试本身并不是一种测试技术，而是一种软件测试风格。这个测试风格，强调测试工程师要同时

开展测试学习、测试设计、测试执行和测试结果评估等一系列的活动，以持续优化测试工作。

其实，在目前的工程实践中，探索式测试发现的缺陷最多，而且发现的缺陷也很有代表性。所以，现在很多企业在探索式测试中都有较大投入。

那么，我现在就和你分享一下什么是探索式测试吧。

什么是探索式测试?

探索式测试，最早是由测试专家 Cem Kaner 博士在 1983 年提出的，并受到当时语境驱动的软件测试学派的支持。后来，Cem Kaner 博士在佛罗里达工学院的同事 James A. Whittaker，凭借着在微软和谷歌担任测试架构师和测试总监的经验积累，撰写了最早的探索式测试书籍（Exploratory Software Testing），扩展了探索式测试的概念和方法。

从本质上来看，探索式测试具有即兴发挥、快速实验、随时调整等特征， Kaner 博士在 2006 年 1 月的 Exploratory Testing Research Summit 会议上将探索式测试的定义总结为以下的一句话（在这里，我直接引用了原文）。接下来，我就和你一起解读一下其中最关键的几个概念吧。

> Exploratory software testing is a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project.

首先，探索式测试是一种软件测试风格，而不是一种具体的软件测试技术。作为一种思维方法，探索式测试强调依据当前语境与上下文选择最合适的测试技术。所以，切记不要将探索式测试误认为是一种测试技术，而应该理解为一种利用各种测试技术“探索”软件潜在缺陷的测试风格。

其次，探索式测试强调独立测试工程师的个人自由和责任，其目的是为了持续优化其工作的价值。测试工程师应该为软件产品负责，充分发挥主观能动性，在整体上持续优化个人和团队的产出。这种思想方法，与精益生产、敏捷软件开发的理念高度一致，这也正是探索式测试受到敏捷团队欢迎的原因之一。

这里需要特别指出的是，探索式测试对个人的能力有很高的依赖：同样的测试风格，由不同的人来具体执行，得到的结果可能会差别巨大。因此，对执行探索式测试的工程师的要求就会比较高，除了要能够从业务上深入理解被测系统外，还要有很强的逻辑分析与推理能力，当然对测试技术以及测试用例设计的融会贯通也是必不可少的技能。

最后，探索式测试建议在整个项目过程中，将测试相关学习、测试设计、测试执行和测试结果解读作为相互支持的活动，并行执行。

注意，这里的并行（run in parallel）并不是真正意义上的并行，而是指对测试学习、测试设计、测试执行和测试分析的快速迭代，即在较短的时间内（比如，1 个小时或者 30 分钟内）快速完成多次循环，以此来不断收集反馈、调整测试、优化价值。这样，在外部看来，就会感觉这些活动是在并行地执行。

这样的理念与敏捷开发中的“小步快跑”、持续反馈的理念不谋而合，因此几乎所有的敏捷团队都会或多或少地应用探索式测试。而且，通常来讲，在敏捷开发的每个迭代中，新开发的功能基本都要依靠探索式测试保证产品的质量。

说到这里，你可能很容易陷入一个误区，那就是探索式测试看起来并没有书面严格意义上的测试设计文档，而且很多测试是在测试执行过程中即兴产生并执行的，那这样的测试风格是不是可以和即兴测试相提并论，两者又有什么区别和联系呢？

探索式测试与即兴测试的区别和联系

虽说探索式测试与即兴测试（Ad-hoc Testing）的风格看起来类似，都是依靠测试工程师的经验和直觉来即兴发挥，快速地试验被测试应用，并不停地调整测试策略。但是，探索式测试相比即兴测试更强调及时“反馈”的重要性。

在探索式测试中，测试工程师不断提出假设，通过测试执行去检验假设，通过解读测试结果证实或推翻假设。在这个迭代过程中，测试工程师不断完善头脑中被测试应用的知识体系，并建立被测应用的模型，然后利用模型、过往经验，以及测试技术驱动进一步的测试。

相比即兴测试完全不注重测试计划和设计，探索式测试要不停地优化测试模型和测试设计。由于测试设计和测试执行的切换速度很快，也就是说切换频率很高，所以会很容易传达“探索式测试没有测试计划和设计”这个错误的信息。然而，实际情况是，探索式测试有明确的测试目标和测试设计，只是测试设计的时间很短，会以很高的频率与测试执行交替切换。

掌握了探索式测试的基本理念之后，接下来我们再简单看一下探索式测试具体是如何实施的。

如何开展探索性测试?

探索式测试也是可以采用分层测试的策略。

通常，**我们首先会对软件的单一功能进行比较细致的探索式测试。**“探索”的过程主要是基于功能需求以及非功能性需求进行扩展和延伸，期间可以采用类似“头脑风暴”的工具，比如 Xmind 等，帮助我们整理思路。

比如，软件系统的用户登录功能就是一个单一的功能，所以作为探索式测试人员，首先应该站在最终用户的角度去理解和使用登录功能。也就是说，探索式测试人员需要了解真正的业务需求，然后基于这些业务需求“探索”软件的功能是否可以满足业务需求。

为此，探索式测试人员需要分析出用户登录功能的所有原子输入项。这里为了简化，假定原子输入项只有用户名、密码和登录按钮。接着，组合这些原子输入项构成最基本典型的测试场景。至于什么才是最基本典型的场景，则取决于探索式测试人员对需求的理解程度。

比如，用真实合法的用户名以及密码完成登录就是一个非常基本典型的场景，如果该场景能够成功登录，就可以切换到下一个；如果该场景不能够成功登录，就需要去“探索”为什么没能登录成功，比如你可能会怀疑是否是因为用户名或者密码是区分大小写的，又或者是不是因为你多次错误的尝试而导致的。

基于你的怀疑，进一步去设计新的测试用例来验证你的猜测。如果你怀疑是用户名或者密码大小写不一致造成的登录失败，那么就需要尝试使用大小写完全一致的用户名和密码进行尝试，之后还应该故意设计一些测试用例采用相同字符但是不同大小写的用户名和密码去做尝试；如果你怀疑登录失败是由于过多次的失败登录引起的，你就应该故意去设计这样的场景来观察系统的实际行为是否符合你的猜想。

总之，通过以上这样的“探索”过程，你就将测试学习、测试设计、测试执行和测试结果评估串联成了一个快速迭代的过程，并在你脑海中快速建立了登录功能的详细模型。

然后，我们往往会开展系统交互的探索式测试，这个过程通常会采用基于反馈的探索式测试方法。基于反馈的探索式测试方法，会运用所有可用的测试技术，以及基于对产品深入理解

后的技术直觉，并结合上一次测试结果的反馈与分析结果，指导测试工程师下一步的测试行动。

还是以用户登录功能为例，在系统交互的探索式测试中，你就不仅要考虑单一的登录功能了，而是要考虑用户登录与系统其他功能相结合的场景。

比如，你可以尝试不登录直接访问登录后的路径去观察系统的行为；再比如，你可以尝试不登录就去查看订单状态的操作等等。这些组合场景的设计主要取决于你想要验证，或者说想要“探索”的系统功能。很多时候这些灵感来自于你之前对系统的探索而取得的系统认识，同时你的技术直觉也在此扮演了重要角色。

最后，我要特别强调一下：其实，很多时候你已经在不知不觉中运用了探索式测试的思想方法，只是你自己并不知道这就是探索式测试而已。所以，探索式测试离你并不遥远，而且实现起来也没想象中的那么难。但是，探索式测试是否可以帮你找出尽可能多的缺陷，还是取决你对系统需求的理解以及根据过往经验而积累的技术直觉。而探索式测试，只是一个测试风格，或者说一个流程方法，所以其本身并不具备任何缺陷发现能力。

总结

通过今天这篇文章，我阐述了一个基本思想是：探索式测试是一种软件测试风格，而不是一种具体的软件测试技术。

作为一种思维方法，探索式测试强调依据当前语境与上下文选择最适合的测试技术，并且强调独立测试工程师的个人自由和责任，其目的是为了持续优化其工作的价值。

探索式测试建议在整个项目中，将测试相关学习、测试设计、测试执行和测试结果解读作为相互支持的活动，并行地执行。

从中我们可以看出，探索式测试的思想方法和精益生产、敏捷软件开发的理念高度一致，这也是探索式测试颇受敏捷团队欢迎的原因之一。

思考题

如果查阅探索式测试相关的技术专著，你会发现探索式测试有很多方法和工具，从中你有哪些收获呢？如果再结合你在实际项目中使用探索式测试的经验，你又有哪些心得体会呢？

欢迎你给我留言，我们一起讨论。

上一篇　42 | 实战：大型全球化电商的测试基础架构设计

下一篇　44 | 测试先行：测试驱动开发(TDD)

精选留言 (8)

写留言

Kerry
2018-10-06

13

我在讲探索式测试，特别是在讲到SBTM中session概念时，会把测试人员比喻作client、把被测对象比作server，这时就可以重新定义"测试"——是测试人员和被测对象之间的一次对话（session），不断质疑系统。为了让学员理解，这时会举"面试"的例子，和茹老师的开头不谋而合，这可以见我去年写的文章: 看家本领之三：软件测试的批判性思维 https://mp.weixin.qq.com/s/SsJyUkIbAkriTaI_CpWJoQ

展开

作者回复: 朱教授来评论，我好紧张！探索性测试更多关注的就是循序渐进，迭代深入的过程，的确是异曲同工的过程。

喵呜呀呵嘿...

2018-10-09

3

我觉得探索性测试最好的地方就是思维没有太多的局限，可以最大程度发散思维地去质疑去测试。我通常会在我测试计划的最后留出一天做这个测试。

作者回复: 非常正确的想法。

才子

2018-11-15

1

老师，探索性测试和错误推断法有什么区别吗?

展开

wanj

2019-06-04

什么时候适合采用探索式测试，探索式测试是否可取代传统的测试方式？采用探索式测试怎么保证功能点不遗漏啊？请茹老师指点

口水窝

2019-05-16

探索式测试，只是一种思维，在工作中已经用到了，而且这种随着经验，不断学习，不同阶段有不同的理解。

小老鼠

2018-11-29

1、探索式测试与随机测试主要区别是有无事后总结，不断调整
2、如何在快速发布版本的环境中执行探索式测试?

作者回复: 探索式测试是测试的思想方法，个人认为和敏捷环境下更需要探索式测试，可以在那个迭代结束前集中开展

Robot_24

2018-11-27

我也想问和错误猜测法有什么区别，老师给上一个提问题的人个回答吧 我看看(●◡●)/

作者回复: 其实没没有很大的必要去严格划分这些概念，关键是要理解其中的设计思路

Joie

2018-10-23

很想在团队内部推行探索性测试

展开