测试专栏特别放送 | 答疑解惑第七期

2018-11-12 茹炳晟

你好，我是茹炳晟。

今天的“答疑解惑”系列文章，我们一起来解决测试新技术、测试人员的互联网架构核心知识这最后两个系列相关的问题。

这期的答疑文章，我不会针对每篇文章后面的思考题展开，而是会选择了四个大家比较关注的问题，和你分享我的观点。如果你的看法不同，或者你还有哪些其他问题的话，欢迎你在这篇文章下面给我留言，我会持续不断地解答你的问题。

当然了，我还是会先用一句话简单概括下每篇文章的内容，并给出对应的链接，方便你复习。

测试新技术系列文章回顾

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。

看到有用户在留言中说到想在实际项目中开展探索式测试，这里我想再给个建议：

高效开展探索式测试的前提是，对被测系统的设计以及行业应用有非常清晰的认识，同时在此基础上以发散的方式对系统可能存在的缺陷进行探索。所以，这就要求测试人员不仅要具有很深的业务领域知识，还需要很强的逻辑推理和分析能力。而这样的人才，属于比较稀缺的。

另外，探索式测试不要到了项目后期再集中展开，而是应该在各个模块级别就尽可能多地去探索，尽量在测试早期就能发现问题。

需要注意的是，一定不要在执行层面，将探索式测试变成了随机测试，你设计的所有后续测试步骤都必须是在你之前的步骤上推演出来的。而且，在执行探索性测试的过程中，你需要明确每个操作的目的是什么，是想证实自己的推论还是要推翻自己的假设。对此，你一定要做到心中有数，否则很容易就会变成无明确目的随机测试。

而从管理层的角度来看，千万不要以探索式测试发现的缺陷数量来考核团队的绩效。因为这样不仅不能提升测试效率，反而会把大量的时间浪费在一些非核心功能的测试上。

在专栏的第 44 篇文章《测试先行：测试驱动开发（TDD）》中，我和你分享了 TDD 的核心思想是：在开发人员实现功能代码前，先设计好测试用例，编写测试代码，然后再针对新增的测试代码来编写产品的功能代码，最终目的是让新增的测试代码能够通过。

正如“叶夏立”在留言中所说，TDD 如何落地才是最核心的问题。所以，我会将这个问题，作为今天这篇文章要回答的第一个问题，和你分享些我的经验。

在专栏的第 45 篇文章《打蛇打七寸：精准测试》中，我通过分析传统软件测试的短板，和你分享了精准测试的概念、必要性、核心思想，以及具体的测试方法。

内容。如果其中有哪些不清楚的问题，我们可以一起探讨，共同进步。

在专栏的第 46 篇文章《安全第一：渗透测试》中，我分享了渗透测试是由专业的安全专家来模拟黑客对系统发起攻击，找到并修复系统的安全漏洞，从而让真正的黑客无机可乘。在这其中，我和你详细分享了渗透测试的知识点，包括常用的测试方法、步骤、工具。

这篇文章更新后，有的用户反馈希望看到实例的演示，这样可以更生动、易于理解，所以这里我决定在今天的第二个问题中，和你分享一个实际的渗透测试实例，满足你的需求。

在专栏的第 47 篇文章《用机器设计测试用例：基于模型的测试》中，我分享了基于模型的测试（MBT）是一种基于被测系统的模型，由工具自动生成测试用例的软件测试技术。这也就决定了，相对于传统软件测试技术来说有优优劣。

所以，我们需要综合考虑项目本身的特点和人员的技术水平，以此决定是否有必要开展 MBT。关于如何判断你的项目是否适合开展 MBT，我决定作为今天的第三个问题，和你展开分享。

测试人员的互联网架构核心知识系列文章回顾

在 48 篇文章《优秀的测试工程师为什么要懂大型网站的架构设计？》中，我主要和你分享了测试人员学习网站架构知识的 why、what、how 的问题，并提出了“由广度到深度”和“自上而下”的架构学习思路，希望可以增强你学习网站架构的信心。

在第 49 篇文章《深入浅出网站高性能架构设计》中，我从测试人员的视角，和你分享了网站的高性能架构设计包括哪些部分，以及在设计测试用例时，需要着重考虑哪些点。而设计到具体的测试方法、工具问题，你可以再回顾一下第 28~34 篇文章（也就是性能测试系列文章）中的相关内容。

在第 50 篇文章《深入浅出网站高可用架构设计》中，我将影响网站高可用的因素归为了三类（即：服务器硬件故障、新应用的发布、应用程序本身的问题），并相应地给出了解决这三类问题的方案。希望这些内容可以帮到你。

在第 51 篇文章《深入浅出网站伸缩性架构设计》中，我和你分享了一个网站的可伸缩性架构设计主要包含的两个层面。其中，一个是指根据功能进行物理分离来实现伸缩，另一个是

在第 52 篇文章《深入浅出网站可扩展性架构设计》中，我和你分享了本专栏的最后一个主题，即网站的可扩展性架构设计。从已有的实现方案来看，实现网站可扩展性架构的主要技术手段包括事件驱动架构和微服务架构。

而在微服务的实现方案中，需要测试人员关注的点，你可以参考第 24 篇文章《紧跟时代步伐：微服务模式下 API 测试要怎么做？》中的相关内容。所以，在这篇文章中，我和你重点分享的是事件驱动架构实现的大致原理，以及测试人员需要额外关注的点。

因为这个系列的文章，更新日期比较近，所以很多用户还没来得及看。所以，我就没有在这篇答疑文章中，设置与这个系列有关的问题。如果你阅读完这个系列的文章，有任何困惑，都可以给我留言，我将和你一起讨论、解决。

问题一：什么样的项目适合 TDD？TDD 如何才能落地？

的确，TDD 这个概念从提出来到现在已经有很长时间了，但实际落地的项目并不多，甚至可以说少之又少。造成 TDD 落地困难的原因有很多，比如很多大型项目本身就不适合做 TDD，TDD 初期阶段的工作划分以及粒度控制都是难点。但我认为最重要的原因是，TDD 需要大幅改变研发团队的流程规范。这种改变在公司层面，尤其是中大型公司是很难实际执行的。

虽然明知落地 TDD 困难重重，但是你又特别想在自己的项目中尝试 TDD，以解决现在的测试方法不能解决的问题。那么，落地 TDD 有哪些可值得借鉴的经验呢？这也正是很多用户关心的，比如昵称为“叶夏立”的用户在文章下面的留言。

叶夏立

写于 2018/10/08

tdd 怎么样做才能落实到项目中，我觉得这才是核心问题，当然不是所有的项目都适合 tdd。不知道茹老师是否能分享一下 tdd 落地推动的做法？

引自：软件测试52讲
44 | 测试先行：测试驱动开发(TDD)

识别二维码打开原文
「极客时间」 App

这里，我根据自己的时间，为你总结了如下几点：

只在一些小型项目，比如前期的 POC 项目中，尝试开展 TDD；

一定要借助 Cucumber 之类的 TDD 或者 BDD 工具，来协助 TDD 的开展；

必须把控好每个测试用例的粒度，不能太大，也不能太小，需要与开发函数以及功能的粒度相匹配；

测试人员必须要有开发背景，否则 TDD 只能是空谈；

必须要得到管理层的大力支持，最好是能自顶向下的推广。

问题二：渗透测试在落地的时候，需要注意哪些问题？

首先说一下，我设立这个问题的初衷，是想通过一个实际的例子，来帮助你理解渗透测试的本质。

所以，我以最常见的 SQL 注入攻击为例，和你简单分享下渗透测试落地时需要主要的问题。假设，我现在要测试用户登录功能，用户登录时会在界面上分别输入用户名（userName）和密码（passWord），然后程序代码会将输入的 userName 和 passWord 填充到下面 SQL 语句中。

复制代码

```
SELECT * FROM users WHERE (name = '" + userName + "') and (pw = '"+ passWord +"');"
```


假设，我们输入的用户名是“Robin”，密码是“12345678”。那么，此时的 SQL 语句如下所示：

复制代码

```
SELECT * FROM users WHERE (name = 'Robin') and (pw = '12345678');"
```

然后，系统就会使用这个 SQL 语句去数据库中查询是否存在该用户，以及该用户的密码是否正确。

此时，如果你是黑客希望通过渗透来非法获取系统信息，你就会尝试设计以下的用户名和密码：

复制代码

```
username 输入 "1' OR '1'='1";
```

这种情况下，用于数据库查询的 SQL 语句就会变成如下所示的样子，就是将 userName 和 password 的部分用 "1’ OR ‘1’ ='1"都代替：

复制代码

```
SELECT * FROM users WHERE (name = '1' OR '1'='1') and (pw = '1' OR '1'='1');"
```

如果你熟悉 SQL 语句的语法，你就会发现黑客查询数据库使用的 SQL 语句，其实和下面这个 SQL 语句是等价的：

复制代码

```
SELECT * FROM users;
```

也就是说，原本用于查询单条用户信息的 SQL 语句已经被黑客改造成了获取全部用户信息的 SQL 语句。这，就是最典型的 SQL 注入攻击手段了。

而我们所讲的渗透测试，就是会去人为模拟这种攻击，以判断系统是否能够成功应对此类攻击。

问题三：如何判断你的项目是否适合采用 MBT，以及你认为会遇到哪些问题可能会阻碍 MBT 的开展呢？

一般来讲，只要系统的设计可以用状态转移图来描述的话，基本都可以采用 MBT。另外，基于 GUI 的系统，因为本身就可以画出页面之间相互跳转的关系图，所以也适合采用 MBT。

很可惜，eBay 内部的项目除了一些实验性的尝试，并没有大规模开展 MBT。但据我所知，业界最近有一家初创企业 AutoTest 正在全力推进 MBT 的落地和应用，而且还发表了很多相关文章，如果你对此感兴趣可以去关注一下。

一个是，探索路径的有效性问题。早期的实施方案，完全基于图论来覆盖可能路径，造成了大量的非法或者不合理的路径。但是，近几年来由于人工智能的介入，大大提升了路径探索的有效性。

另一个是，如果只是用 MBT 完成单纯测试的话，收益比会比较低。只有将 MBT 和自动化测试结合在一起，才能发挥 MBT 的优势。在这方面，eBay 一直在尝试，试图将 Selenium 和 MBT 集成到一起，目前已经有了初步成果，实现了用模型导出实际可以执行的自动化测试用例的 POC。

问题四：测试工程师如何应对面试？

首先，我并不鼓励为了应对面试，而去做特别的准备，你还是应该在平常的工作中多积累。**面试的过程，本来就是尽可能地反映你真实的技术水平以及业务熟练程度的交流，关注的重点应该是如何将你自己掌握的技术和知识更好地展现出来，而不是把你不懂的知识包装成你已经“懂”的知识。**为此，我有如下几点小建议供你参考：

当被问及相关测试工具的时候，除非问到了该工具使用上的细节，否则尽可能避免谈及工具使用的细节，而是应该更多地阐述工具本身的原理、和同类工具相比的优劣势，以及这个工具可以以什么样的方式帮你解决问题。这时，你的视野一定要高，不能局限于细节。当然，这也就要求你能够充分理解这个工具的原理和使用方法。

当被问及特定算法实现的时候，刚开始的时候，一定不要试图去寻求最优解法，而是要考虑最基本的实现，然后在此基础上迭代优化。因为，优秀的面试官希望看到的不是最优解，而是你解决问题的过程，以及在这个迭代过程中的逻辑推理。

当被问及你所不熟悉的测试技术时，如实回答就好，不要试图去掩饰。很多时候面试官看重的并不是你知不知道，而是当你不知道的时候你会怎么做。

最后，感谢你能认真阅读第 43~52 这 10 篇文章的内容，并写下了你的想法和问题。期待你能继续关注我的专栏，继续留下你对文章内容的思考，我也在一直关注着你的留言、你的学习情况。

咱们的答疑环节暂告一段落了，但这并不意味着结束。如果你在学习过程中遇到了什么问题，还可以继续给我留言，我会持续不断地回答你的问题。

上一篇　测试专栏特别放送 | 答疑解惑第六期

下一篇　测试专栏特别放送 | 浅谈全链路压测

精选留言 (3)

写留言

萨拉热窝的...
2019-01-10

老师，开发平台这种怎么测试？开发人员用这个平台做开发，，，但是这个平台本身怎么测？

作者回复: 这个不是一两句话可以说清楚的，这就像hp自己怎么来测试自己的qtp和loadrunner一样，基本思路还是要覆盖各种可能的使用场景，当然这类平台测试的成本往往很高

胜杰
2018-11-19

2、根据bug的优先级别，报给开发去修，优先级高的的bug不修的话则测试不通过不予发布；
3、软件测试工程师的职责就是确保软件在发布前尽可能的发现问题解决问题

展开

作者回复: 感谢鼎力支持

→_→晓^O^

2018-11-15

作为一名测试员发现了问题，如何更好的推动BUG被解决呢？有时候推不动，就很累，一天天也感觉过的很不充实。茹老师，这方面可以指导下吗？