

测试专栏特别放送 | 答疑解惑第三期

2018-11-02 茹炳晟

软件测试52讲

[进入课程 >](#)



你好，我是茹炳晟。

今天这篇文章是“答疑解惑”系列文章的第三期，在这一期里面我做了个小小的改变，不再是针对每一篇文章后面的思考题，以及“你”的留言，来展开分享。这次，我选择将这个专栏的第二个系列：GUI 自动化测试系列，作为一个整体，回答下你的问题。

根据目前 GUI 测试技术的应用情况，以及“你”在留言中留下的期望进一步了解的问题，也为了能够讲清、讲透，我还是选择了五个问题，为你解答。当然了，如果你关注的问题并未涵盖其中的话，也不要遗憾，你可以在这篇文章下继续给我留言，我会持续关注并回答你的问题。

接下来，我还是会先为你总结一下，GUI 自动化测试系列的全部 10 篇文章的内容，并为你提供文章链接，希望你可以通过回顾旧文获得新知。

系列文章内容回顾

传达给你的内容是，基于 Selenium 搭建 GUI 自动化测试用例的方法，以及 Selenium 1.0、2.0 的实现原理。所以，对于具体的用例设计方面的细节，比如构建一个 GUI 测试用例时需要封装哪些类等，我并没有详细展开。

也正如 “Cynthia” 在留言中所说的，一些资料会直接带你分析源码，但不会为你讲述工具的原理，所以虽然你会用这个工具，但是类似于为啥 Chrome 可以跑的 case、Firefox 跑不了、为啥 Web Driver 还要一个浏览器装一个这种问题，你永远不会明白。反馈到具体的测试工作上，就会出现知道要这么做，但是不知道为什么要这么做的尴尬局面。

在专栏的第 13 篇文章 [《效率为王：脚本与数据的解耦 + Page Object 模型》](#) 中，我和你分享了什么是数据驱动测试，让你明白了“测试脚本和数据解耦”的实现方式以及应用场景；然后，我从 GUI 自动化测试历史发展演变的角度，引出了 GUI 测试中的“页面对象模型”的概念。

在这篇文章最后，我希望你思考一下“是否应该在页面对象模型中封装控件的操作”这个问题。我看到留言区的回复，也都很精彩，可谓是思维清晰、逻辑严谨。这里，我想再阐述一下我的观点，详情请见问题二。

在专栏的第 14 篇文章 [《更接近业务的抽象：让自动化测试脚本更好地描述业务》](#) 中，我以“如何把控操作函数的粒度”和“如何衔接两个操作函数之间的页面”这两个问题为引子，和你分享了业务流程的概念、核心思想和适用的场景。

说实话，看到这篇文章下面的留言我很感动，为什么呢？因为我分享的测试思想得到了你的认可，也吸引你围绕着这一个主题展开了讨论。在这些留言中，我也仿佛看到了自己当年为了做好某一项测试而尝试的一系列方案。所以，在此我也想对你道声感谢。

在专栏的第 15 篇文章 [《过不了的坎：聊聊 GUI 自动化过程中的测试数据》](#) 中，我从创建测试数据的技术手段和时机两个方面，和你分享了在实际项目中，需要综合运用 API 调用和数据库操作来创建测试数据，并根据测试数据自身的特点，分而治之地采用 On-the-fly 和 Out-of-box 的方式，以寻求数据稳定性和数据准备效率之间的最佳平衡。

因为这个专栏后面有一个单独的系列去讲测试数据准备的那些事儿了，所以今天我不再过多地展开这个问题了。当然，你也可以在这篇文章下面，留下你关注的与测试数据相关的

在专栏的第 16 篇文章 [《脑洞大开：GUI 测试还能这么玩（Page Code Gen + Data Gen + Headless）？》](#) 中，我一下和你分享了页面对象自动生成、测试数据自动生成、无头浏览器这三种技术。或许，在你看来有些停留在概念层面了。

说到为什么我没有针对某一个具体的概念，展开分享，其实我是这么考虑的：页面对象自动生成，主要用到的就是 QTP 这个工具，而对这个工具的使用，你可以通过它的文档来轻松上手，并且这个工具的内部实现也没有对外公开，所以这部分内容完全从工具使用的角度来讲的话，并不能发挥这个专栏的价值。同样地，测试数据自动生成、无头浏览器也是这个道理。我还是希望你能在知道了这些概念之后，可以结合自己的实际工作，动手实践，这样的效果才是最好的。

所以，今天我也就不再过多的展开这个话题了。如果你在落地这三个概念的时候有任何问题，也可以给我留言，希望可以帮到你。

在专栏的第 17 篇文章 [《精益求精：聊聊提高 GUI 测试稳定性的关键技术》](#) 中，我为你归纳了五种造成 GUI 自动化测试不稳定的主要因素，即：非预计的弹出对话框、页面控件属性的细微变化、被测系统的 A/B 测试、随机的页面延迟造成控件识别失败，以及测试数据问题。然后，我针对每种问题，给出了对应的解决思路。

当时为了不淹没提高 GUI 测试稳定性的关键技术这个主题，我并没有通过实例展开这五个因素及其对应的解决方案。今天为了加深你的理解，我决定针对随机失败的重试（retry）和你分享一个实例，作为今天我要回答的第三个问题。

在专栏的第 18 篇文章 [《眼前一亮：带你玩转 GUI 自动化的测试报告》](#) 中，我和你分享了一份理想的 GUI 自动化测试报告应该包括哪些内容，并和你分享了 eBay 在全球化 GUI 测试报告中的创新设计。希望这些方法，你也可以经过适当改进，用到自己的测试项目中。

有些读者感觉这篇文章不够过瘾，想要知道某一个技术细节的实现。所以，这里我会针对基于 GUI 测试报告，和你再分享一个小例子，和你说说故事板样式的 GUI 测试报告具体是如何实现的，也就是后面我要分享的第四个问题。

但是，我还是想要再强调一下，这篇文章的设计初衷并不是去解释每种报告的实现细节，而是帮助你从多元化的视角去思考一个高效的测试报告应该是什么样的。

望可以帮你解决测试策略如何设计、测试用例脚本如何组织这两个问题。

其实，通过这篇文章，我已经和你说清楚了大型项目中的 GUI 自动化测试策略这个问题，所以在今天这篇文章中我就不再继续展开这个话题了。

在专栏的第 20 篇文章[《与时俱进：浅谈移动应用测试方法与思路》](#)中，我和你分享了 Web App、Native App 和 Hybrid App，这三类移动应用的测试方法，以及移动专项测试的思路与方法。其实，这三类移动 App 的测试，和 GUI 自动化测试的思想是相通的，只不过是针对移动应用自身的特点，又有了一些独特的方法而已。

在专栏的第 21 篇文章[《移动测试神器：带你玩转 Appium》](#)中，我用 Appium 手把手地带你实现了一个移动应用的测试用例，并本着知其所以然的原则，和你分享了 Appium 的实现原理，希望借此可以带你玩转 Appium，完成自己的移动应用测试。

这两篇关于移动 App 测试的文章更新后，大家留言都很踊跃，也可以反映出大家很关注这块内容。所以，今天我再选择微信小程序的自动化测试，和你再分享下业界的主流工具，以此作为今天的最后一个问题。

问题一：目前互联网企业在 GUI 测试上的投入在不断减少，而为什么我要花这么多精力去讲 GUI 测试？

这个问题，很多人都想知道答案。那么我就说说这么设计的考量吧。

当前互联网产品的测试重点的确逐渐在向后端（API）迁移，GUI 自动化测试所占比重越来越低，出现这种情况的原因主要有两个：

GUI 自动化测试用例的 ROI 一直上不去，投入较大。一方面，测试用例本身还很难做到 100% 的稳定；另一方面，被测系统界面发生变化时，测试用例的维护成本一直高居不下。

微服务架构的普及使得后端逻辑可以被重用到大量的前端界面中，比如同一个后端 API 既可以通过 PC 端的浏览器请求来访问，也可以通过移动端 Native App 的请求来访问，此时保证后端 API 的质量就成了保证前端质量的基础。也是因为这个原因，API 测试取代了部分 GUI 自动化测试。

实际使用的 GUI 做过测试，你是否敢直接发布上线。

很多时候，后端 API 的功能都正常并不能保证前端的 GUI 功能测试就一定没有问题。所以，现在很多互联网企业依然会在 GUI 自动化测试上做投入，只不过是将 GUI 自动化测试用例的设计原则从原本“全面覆盖”，进化成为了“只覆盖最基本的核心业务功能”。

问题二：是否应该在页面对象模型中封装控件的操作？

这其实是个测试架构设计的问题，而且一直以来都有争议。

如果在页面对象中封装控件的操作，实现起来相对简单，而且符合传统的面向对象的设计思想，即：一个对象既有对象描述又有对象操作。

但是，从测试架构的层次上来看，这种设计不够规整，而且将对象的识别和操作进行了耦合。而这种耦合就意味着需要对对象的识别和操作一同进行更新和版本管理，但现实情况是对象的识别需要经常更新，而对象的操作相对来说很稳定，这种不必要的依赖关系又增加了测试的维护成本。

如果将页面对象和对象的操作分开封装，即：让对象操作的类依赖于对象定义的类。这种设计，在测试架构的层次上来看比较清晰，而且可以分别来实现页面对象识别和对象操作的版本管理。同时，如果你已经采用了页面对象自动生成技术，就可以直接生成页面对象类，从而保证页面操作类的稳定。这样做的好处也很明显，即能够以自动化的方式来降低由页面变化引起的改动工作量。

由此可见，这两种方式各有优劣。那么，根据我的经验来看，对于初级阶段的 GUI 自动化测试，直接在页面对象中封装控件操作的方式更简单直接；而随着 GUI 自动化测试不断成熟，在需要考虑引入页面对象的版本管理以及页面对象的自动生成的时候，将页面对象和对象的操作分开封装的方式，就是大势所趋了。

问题三：如何通过重试（retry）机制，来应对由随机的页面延迟造成的控件识别失败？

正如我在第 17 篇文章 [《精益求精：聊聊提高 GUI 测试稳定性的关键技术》](#) 中提到的，重试可以是步骤界别的，也可以是页面级别的，甚至可以是业务流程级别的。

函数上封装重试逻辑，而且默认情况下，我建议你可以启用这个级别的重试。

再往上一层是业务流程，也就是 Business Flow 的重试。如果一个业务操作失败了，我们就可以考虑重新执行这个业务流程。

但是，这其中需要特别注意的是，业务流程对数据本身可能存在依赖，所以业务流程重试的合理性需要根据测试用例的上下文来决定，尤其要关注测试数据是否可以支持可重复执行。

一般来讲，我们会在自动化测试框架中实现 Business Flow 的重试。但是，我建议默认关闭该功能，只在你确定需要进行业务流程级别的重试时才显式启用 Business Flow 层面的重试。这么做的主要原因是，业务流程的重试会涉及测试数据的问题，只有当你非常肯定业务流程的重试不受测试数据的影响时，我们才会人为启用业务流程级别的重试。

最上面一层是测试用例级别的重试。当一个测试用例执行失败的时候，我们可以考虑重新执行整个测试用例，一来希望排除随机出错的可能，二来是评估出错的步骤是否可以重现。

这种测试用例级别的重试，一般是由 CI 流水线完成的，而不是由自动化测试框架实现的。CI 流水线会整理出所有失败的测试用例列表，然后在 CI 流水线脚本中发起重试。

问题四：故事板样式的 GUI 测试报告，具体是如何实现的？

故事板样式的 GUI 测试报告，指的是按时序对 GUI 操作界面进行截图生成的测试报告。对于这类报告的实现主要考虑两个方面的因素：

一是，获取屏幕截图。这一步，我们可以通过封装控件操作函数来实现，比如在自行封装的 click 函数中，先调用截图函数然后再调用真正的 click 操作。

二是，截图在报告中的展现形式。这一步，我们可以直接使用现成的 HTML5 的 PPT 框架，这样我们只要按照框架要求来生成屏幕截图的数据结构即可，而无需去关注复杂的 HTML5 的处理逻辑。

这里，我强烈推荐 reveal.js。eBay 就是用这个框架开发了自己的故事板样式的 GUI 测试报告。其中，在第 18 篇文章[《眼前一亮：带你玩转 GUI 自动化的测试报告》](#)中的测试报

问题五：关于微信小程序的自动化测试，以及用到的测试工具。

很多读者在留言中问到了微信小程序的测试问题，这里我稍微展开下。

其实，微信小程序的测试，在测试设计和执行思路同移动端测试并没任何区别。那为什么很多人会问到微信小程序的测试呢？

我觉得主要原因是，在实际开展自动化测试的时候，由于缺少专业的、有效的工具，使得微信小程序的测试在我们看来难以执行。

其实呢，有一款来自腾讯的微信小程序测试工具 XTest 就是个不错的选择，可以很方便地支持测试用例的录制。这里有一篇使用这个工具进行实际[小程序测试的文章](#)，你可以从中获得一些使用细节。

最后，感谢你能认真阅读第 12~21 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。

感谢你的支持，我们下一期答疑文章再见！



软件测试52讲

从小工到专家的实战心法

茹炳晟

eBay中国研发中心
测试基础架构技术主管



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

上一篇 52 | 深入浅出网站可扩展性架构设计

下一篇 测试专栏特别放送 | 答疑解惑第四期

精选留言 (6)

写留言



口水窝

2019-06-05



重读一遍，还是尤为尽然，记录笔记，制定自己的学习计划。

展开 ∨



johnny

2019-04-18



GUI自动化可以发现一些功能上的问题，但对于多浏览器页面布局方面的问题。比如针对移动应用“Web App”，由于移动应用需要使用移动浏览器访问。那我测试其中的某个页面在各种浏览器下（qq浏览器、UC浏览器、safari浏览器）显示效果是否一致、是否和设计稿一致、样式显示有问题，在GUI自动化中实现思路是怎样的？

展开 ∨



春明

2019-03-05



老师，对于如何连接两个不同页面的做法还不是很清楚。能具体举个例子吗？

展开 ∨



仰望星空

2018-11-04



老师，能不能讲讲cucumber测试框架的应用呢

展开 ∨



诺漫谛小金...

2018-11-03





下载APP



Robert小七

2018-11-02



Gui测试为了使用例稳定性提高，每一个测试用例都需要登陆退出吗？

展开 ∨