

19 | 真实的战场：如何在大型项目中设计GUI自动化测试策略

2018-08-10 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 12:30 大小 5.73M



在前面的文章中，我介绍过 GUI 自动化测试的页面对象模型和业务流程封装等相关知识，也提到过大型全球化电商网站的 GUI 自动化测试，那如何把已经学到的 GUI 测试理论知识用到大型全球化电商网站的测试中呢？

今天，我的分享就从“实战”这个角度展开，带你看看实际的大型全球化电商网站的 GUI 自动化测试如何开展。这场实战，我将从以下两个方面展开：

1. 测试策略如何设计？这一点，我会根据亲身经历的 actual 项目，和你探讨 GUI 测试的分层测试策略。
2. 测试用例脚本如何组织？需要注意的是，对于这个问题，我不是要和你讨论测试用例的管理，而是要讨论测试用脚本的管理。比如，当需要组装上层的端到端（E2E）测试时，如何才能最大程度地重用已有的页面对象以及业务流程（business flow）。

如果你所在的企业或者项目正在大规模开展 GUI 测试，并且准备使用页面对象模型以及业务流程封装等最佳实践的话，那么，你很可能会遇到本文所描述的问题并且迫切需要相应的解决办法。

大型全球化电商网站的前端模块划分

在正式讨论大型全球化电商网站的 GUI 自动化测试策略设计之前，我先简单介绍一下电商网站的前端架构，为避免过多的技术细节引起不必要的干扰，我只会概要性地介绍与 GUI 自动化测试密切相关的部分。

由于大型全球化电商网站的业务极其庞大，所以前端架构也要按照不同的业务模块来划分，比如用户管理模块、商户订单管理模块、商户商品管理模块等等。

当然由于这些前端模块都会使用项目自己封装的组件库，比如自定义开发的列表组件、登录组件、信用卡组件等，我们通常会把自定义开发的这些所有组件都放在一个“公共组件库”中，为前端模块提供依赖。

所以，从代码库（Repository）的角度来看，各个前端模块都有各自独立的代码库，除此之外还会有一个公共组件的代码库。

大型全球化电商网站的 GUI 自动化测试策略设计

了解了大型全球化电商网站前端模块的划分后，我们再来看看它的 GUI 自动化测试策略是如何设计的。

总体来看，对大型网站来讲，GUI 自动化测试往往应该做得比较轻量级，而不应该把大量的功能测试，以及功能的组合测试放在 GUI 自动化测试中，正如我在第 11 篇文章[《互联网产品的测试策略应该如何设计？》](#)中谈到的，GUI 测试通常只覆盖最核心且直接影响主营业务流程的 E2E 场景。

但同时，GUI 的验证一定不是在系统全部完成后才真正开展的，也应该是分阶段、分层次来设计制定测试策略的，那么接下来我也将会按照自底向上的顺序分层次介绍 GUI 自动化的测试策略。

首先，要从前端组件的级别来保证质量，也就是需要对那些自定义开发的组件进行完整全面的测试。

公共组件库会被很多上层的前端模块依赖，它的质量将直接影响这些上层模块的质量，所以我们往往会对这些公共组件进行严格的单元测试。最常用的方案是：基于 Jest 开展单元测试，并考量 JavaScript 的代码覆盖率指标。

Jest 是由 Facebook 发布的，是一个基于 Jasmine 的开源 JavaScript 单元测试框架，是目前主流的 JavaScript 单元测试方案。

完成单元测试后，往往还会基于被测控件构建专用的测试页面，在页面层面再次验证控件相关的功能和状态。这部分测试工作也需要采用自动化的形式实现，具体的做法是：

1. 先构建一个空页面，并加入被测控件，由此可以构建出一个包含被测控件的测试页面，这个页面往往被称为 Dummy Page；
2. 从黑盒的角度出发，在这个测试页面上通过手工和自动化的方式操作被测控件，并验证其功能的正确性。

对于自动化的部分，需要基于 GUI 自动化测试框架开发对应的测试用例。这些测试用例，往往采用和 GUI E2E 一样的测试框架，也是从黑盒的角度来对被测控件做功能验证。

其次，每一个前端模块，都会构建自己的页面对象库，并且在此基础上封装开发自己的业务流程脚本。这些业务流程的脚本，可以组装成每个前端模块的测试用例。

以用户管理模块为例，测试用例的组装过程如下：

首先，把用户管理模块中涉及到的所有页面，比如登录页面、用户注册页面等，按照页面对象模型的要求写成 Page 类；

然后，利用这些 Page 类封装业务流程脚本，比如用户登录流程，用户注册流程等；

最后，在 GUI 测试用例脚本中，调用封装好的业务流程脚本构成该模块的 GUI 测试用例。

在这个阶段，测试用例需要完整覆盖该模块的所有业务逻辑以及相关的功能测试点，但是并不会实现所有测试用例的自动化。

自动化测试用例的原则，通常是：优先选取业务关键路径以及 Happy Path 作为自动化测试的范围。在资源充裕的情况下，我们希望这个阶段的自动化率可以达到 70%-80%。所以，前端模块的质量保证主要依赖这部分测试。

如果你比较细心，一定还记得我在之前的文章中有提到过，“GUI 的自动化测试往往只覆盖最核心且直接影响主营业务流程的 E2E 场景”，并且“GUI 测试遵循“手工测试为主，自动化为辅”的策略，而这里又建议说理想的自动化率应该达到 70%~80%，是不是有点前后矛盾的感觉。

其实，这是两个层面的测试，这里 70%-80% 的 GUI 自动化覆盖率是针对模块级别的要求；而“自动化测试为辅，手工为主，以及只覆盖核心业务场景”针对的是系统级别的 E2E 测试。这里容易引起混淆的点是模块测试和系统级别 E2E 测试都是属于 GUI 自动化测试的范畴。

最后，组合各个前端模块，并站在终端用户的视角，以黑盒的方式使用网站的端到端 (E2E) 测试。这部分的测试主要分为两大部分：

一部分是，通过探索式测试的方法手工执行测试，目标是尽可能多地发现新问题；

另一部分是，通过 GUI 自动化测试执行基本业务功能的回归测试，保证网站核心业务相关的所有功能的正确性。

虽然这部分端到端 GUI 测试用例的绝对数量不多，往往是几百个的规模，但是对于保证最终网站的质量却起着非常关键的作用。

可以这样说，如果这些端到端的 GUI 自动化测试用例 100% 通过，那么上线后基本业务功能的质量就不会有大问题。所以，这部分测试工作的重要性不言而喻。

那么，接下来的问题是，应该由谁来开发这部分端到端的 GUI 自动化测试用例呢？

每个前端模块都会有对应的 Scrum 团队，他们会负责开发该模块的页面对象模型、业务流程脚本以及测试用例。而端到端的 GUI 自动化测试不隶属于任何一个 Scrum 团队。

这种情况下，最好的做法就是：成立一个专门的测试团队，负责这种系统级别的 GUI 测试。这样的团队，往往被称为 E2E 测试团队。

很显然，如果由 E2E 团队从无到有地开发这部分 GUI 自动化测试的脚本，效率低下。而且，这部分测试会涉及很多前端模块，当各个前端模块的需求、业务流程以及页面实现有任何变动时，E2E 团队都很难做到及时更新。

所以，解决这个问题的最佳实践就是：E2E 团队应该尽可能地利用各个模块已有的页面对象和业务流程脚本，组装端到端的 GUI 测试。

这样一方面最大程度地减少了重复工作，另一方面可以把各个模块的变更及时反映到端到端的 GUI 测试中，因为端到端的 GUI 测试用例是直接调用各个模块的页面对象和业务流程脚本，而这些页面对象和业务流程脚本都是由每个模块自己的 Scrum 团队维护的。

而为了能够在端到端的 GUI 自动化测试中，复用各个模块的页面对象和业务流程脚本，我们就必须考虑的问题，也就是我今天要和你探讨的第二个话题：GUI 自动化测试脚本应该如何组织？

大型全球化电商网站的 GUI 自动化测试脚本管理

原有的方案，不能解决端到端的 GUI 自动化测试复用各个模块的页面对象和业务流程脚本的问题，在不断的实践中，我总结了一个如图 1 所示的脚本组织结构来解决这个问题。

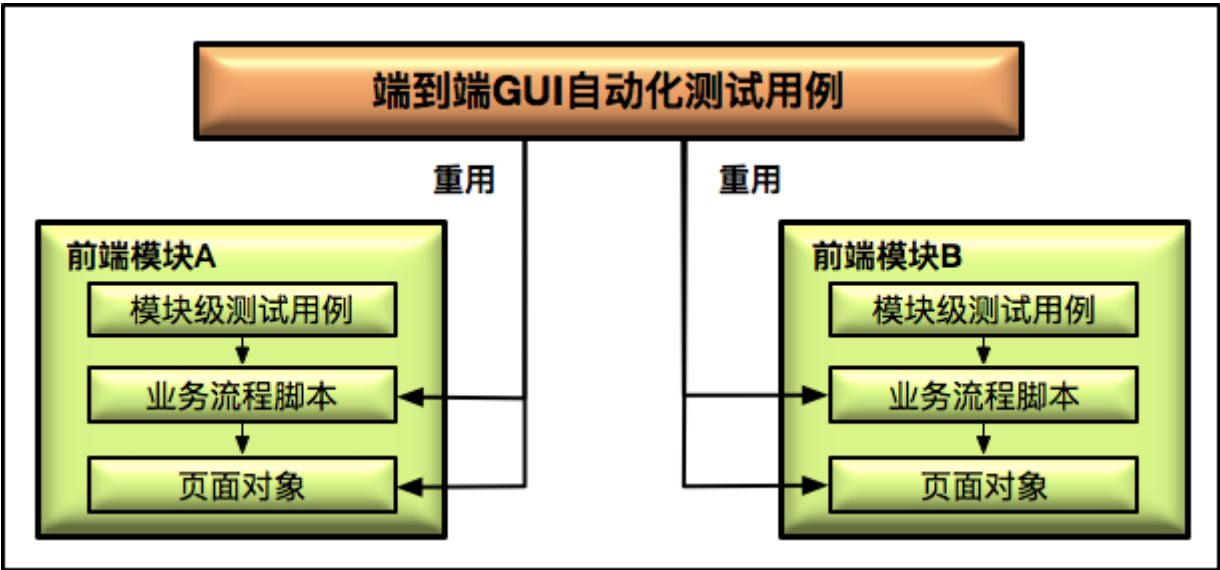


图 1 大型全球化电商网站的 GUI 自动化测试脚本管理

也就是说，将各个模块的页面对象和业务流程脚本放在各自的代码库中，并引入页面对象和业务流程脚本的版本管理机制，通常采用页面对象和业务流程脚本的版本号和开发版本号保持一致的方案。

比如模块 A 的版本号是 V1.0.0，那么对应的页面对象库和业务流程脚本的版本号也应该是 V1.0.0。

在端到端的 GUI 自动化测试脚本中，引用各个模块正确的页面对象和业务流程脚本的版本号，测试用例代码就可以直接调用模块的页面对象和业务流程脚本了。

具体在测试项目中，模块版本的依赖往往是用 POM 来配置的，如图 2 展示了一个典型测试项目的 POM 文件中的版本依赖关系，其中引用了两个模块，appcommon 模块对应的就是上文提到的“公共组件库”，而 app.buy 对应的就是具体依赖的前端模块。

由于这只是一个示例，所以我只保留了两个依赖模块，实际的端到端 GUI 测试项目往往会包含大量的模块依赖。

```
1  <properties>
2      <com.demoproject.appcommon>8.0.1</com.demoproject.appcommon>
3      <com.demoproject.app.buy>8.6.1</com.demoproject.app.buy>
4  </properties>
5
6  <dependencyManagement>
7      <dependencies>
8          <dependency>
9              <groupId>com.demoproject.appcommon</groupId>
10             <artifactId>com.demoproject.appcommon.uiautomation.flow</artifactId>
11             <version>${com.demoproject.appcommon}</version>
12         </dependency>
13         <dependency>
14             <groupId>com.demoproject.appcommon</groupId>
15             <artifactId>com.demoproject.appcommon.uiautomation.page</artifactId>
16             <version>${com.demoproject.appcommon}</version>
17         </dependency>
18         <dependency>
19             <groupId>com.demoproject.app.buy</groupId>
20             <artifactId>com.demoproject.app.buy.uiautomation.flow</artifactId>
21             <version>${com.demoproject.app.buy}</version>
22         </dependency>
23         <dependency>
24             <groupId>com.demoproject.app.buy</groupId>
25             <artifactId>com.demoproject.app.buy.uiautomation.page</artifactId>
26             <version>${com.demoproject.app.buy}</version>
27         </dependency>
28     </dependencies>
29 </dependencyManagement>
```

图 2 典型测试项目的 POM 文件中的版本依赖关系

在这种管理机制下，E2E 团队不需要重复开发任何的页面对象和业务流程脚本，而且可以始终保证与各个模块的最新实现同步，同时端到端的 GUI 测试用例脚本也会比较稳定，不会因为各个模块的改动而频繁地修改。

这样一来，E2E 团队就会有更多的时间和精力去设计并执行探索式测试，发现更多的潜在缺陷，形成良性循环。

总结

我从实战的角度，介绍了大型全球化电商网站 GUI 测试的策略设计以及测试脚本管理的问题：

首先，要从前端组件的级别来保证质量，也就是需要对那些自定义开发的组件进行完整全面的测试。通常前端组件会基于 Jest 做比较严格的单元测试。

其次，每一个前端模块，都会构建自己的页面对象库，并且在此基础上封装开发自己的业务流程脚本。这些业务流程的脚本，可以组装成每个前端模块的测试用例。

最后，把各个前端模块组合在一起之后，站在终端用户的视角以黑盒的方式使用网站的端到端的测试。端到端的测试应该尽可能多地重用各个模块的页面对象库和业务流程脚本来完成。

而为了能够在端到端的 GUI 自动化测试中，复用各个模块的页面对象和业务流程脚本，我建议的方案是：对各个前端业务模块的页面对象库和业务流程脚本，实施版本化管理机制。

思考题

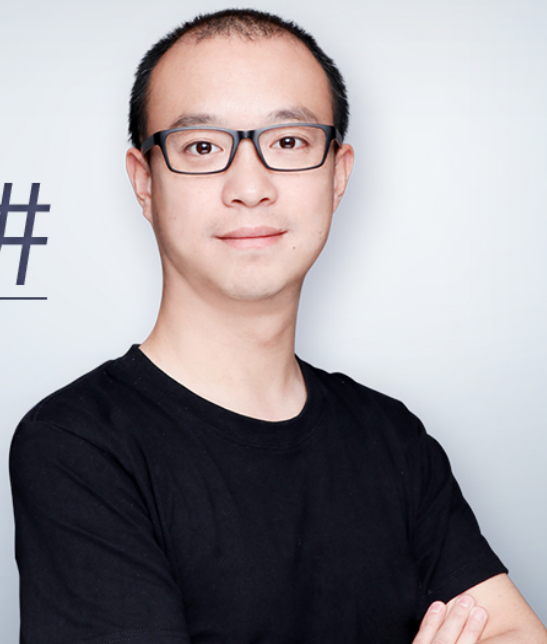
你所在的公司或者项目团队，是否已经或者正计划开展 E2E GUI 测试？开展过程中，遇到过什么难题，你们又是如何解决的？

欢迎你给我留言。

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。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 18 | 眼前一亮：带你玩转GUI自动化的测试报告

下一篇 20 | 与时俱进：浅谈移动应用测试方法与思路

精选留言 (10)

 写留言



arthur

2018-08-19

 4

茹老师您好，我想请教几个问题：

1. 你们团队会写手工测试用例吗？如果写，是怎么写的呢？如果不写，回归测试完全自动化覆盖吗？还是只能熟悉功能的同学进行探索测试？
2. 你们测试团队人员组成是什么样的呢？测试分功能测试组，自动化测试组，E2E组，性能，安全测试组，还是测试组内的人员所有事情都需要做？...

展开 ▾



hohofugao

2018-08-12

 2

用pom维护不同案例的依赖关系，但是怎么保证不同案例数据前后一致？比如登陆用户用

A, 后面注销用户如果数据是B呢



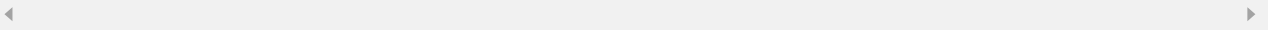
syland215

2018-08-10

👍 2

1. 理论上完全符合 Unit > Integration > E2E 的理论模型，而且很好的应用到实践，赞，不知道这套实现的投入产出比如何？目前使用过程中是否出现过什么问题？是否还有可以改进的地方？
2. 对于持续迭代的大型 web 站，看起来这套实现已经被实践验证，可以推广使用，但是...
展开 ▾

作者回复: 关于第一个问题，我是里面还会有很多问题，比如有些功能会同时修改多个前端模块，如何保证所有模块都升到最新版本，再比如必须要规范化各个模块的log输出，否则e2e的时候出错到底如何高效定位是哪个模块的问题... ROI的话主要看项目规模，当项目比较大的时候，尤其还需要支持全球化gui的时候，这种方式是唯一的出路。对于小项目就看自己的情况了。第二个问题其实还是要看你的项目周期，是长期的还是短期的，如果是那种一次性的项目，压根没必要自动化，如果会长期维护的话，还是建议要去做做的



丹丹兒 ©

2018-08-16

👍 1

GUI 的 E2E，应该可以把平时积累的手工冒烟测试用例转化为自动化测试
展开 ▾



klxiaoqi

2018-08-15

👍 1

老师后面可不可以出python示例
展开 ▾



lerame

2019-04-29

👍

老师，可以这么理解吗：模块就是指注册、登录、加入购物车、付费这种。而e2e是指一个用户注册——登录——浏览——加入购物车——付费这一整个流程的任何正向或异常操作？另外，happy path具体怎么理解，是指容易实现的流程路径吗？



小老鼠

2018-10-25



1、ebay是优化过的selenium 进行GUI测试还是用UTF?
2、你经常说大型商务网站一天两个版本。这里又说通过版本号来管理。请问一天两个版本有专门的版本号吗？若有，与自动化测试脚本通过版本号管理有关系吗？



涅槃Ls

2018-08-14



打卡19，加油

展开 ∨



图·美克尔

2018-08-13



老师你说不偏向于任何语言...但我就看到基于Java的实现方式...请问版本依赖管理在python中如何实现呢



on the w...

2018-08-10



目前公司还没有实行E2E GUI 测试

展开 ∨

作者回复: e2e是上线前最后一道防线，还是很有必要做的，哪怕不是自动化的方式

