

13 你真的了解自动化测试么?

更新时间: 2019-09-19 15:44:06



世上无难事,只要肯登攀。——毛泽东

通过前边对互联网行业的了解,我们真真切切的认识到自动化测试,对于我们来说非常非常重要。我相信很多同学已经了解甚至编写过自动化测试脚本了,但是请你静下心来想这样一个问题:自动化测试到底是什么?

举一个形象一点的例子,我们俩在同一个公司且座位相邻,这天你跟我说:风落啊,我午休时候很容易睡过头,你能不能每天下午一点时候拍醒我?像我这么善良,又有同情心的人一定是满口答应了,可是一周之后我觉得叫你起床这件事又重复,而且还要每天记得,比较累。怎么办呢?我想了个好办法,花一周时间做了一个机器人,这个机器人每天1点固定去你座位上把你打醒。这样我就可以去做其他事情了,解放了我自己的人力来提升效率。

这就是最简单的自动化测试了。归纳成比较官方的表达,就是把人为的测试行为转化为机器执行。

三 优秀测试工程师的必备思维39讲 / 13 你真的了解自动化测试么?

进自动化测试的过程中，会产生很多的错误理解，这时候，我们再来想另一个问题：你做自动化测试的预期是什么？或者换个方式：你的项目中，进行自动化测试的目标是什么？

我也就这个问题问过很多同学，我一点都不意外的收获了很多这样的回答：我不知道为什么要做自动化测试，领导要做，我便做了。

正是源于根本不清楚自动化测试的目的，也便会漫无目的去进行投入，所以很多公司的自动化测试走上了一条弯路。我曾经去到一个，自称自动化测试覆盖率达到 95% 的公司去做相关咨询，发现他们的自动化是这样的：

- 按照功能测试用例用代码实现
- 通过 Eclipse，选好环境
- 右键 Run As TestNG
- 一个人，甚至是两个人一起目不转睛的盯着显示器
- 步骤执行不成功，回到代码，Debug 找问题，修改代码
- 部分脚本执行过程中卡住了，手动辅助点击，继续执行

看到这里，我几乎崩溃了，这样真的是为了自动化而自动化，即使覆盖率达到了 95%，也并没有真正提升他们的效率，而这，正是他们要我去帮助解决的问题。很无奈的我，抓了一个自动化测试工程师，问他们这样做的原因，回答更是让人吐血：老板要做自动化，自动化不就是写脚本么，这难道不是自动化么？我们觉得这样很好啊。为什么要做自动化，这是领导的决定，我们都是执行的。而且以前都是这么做的嘛，也没人说不好啊？

这样的灵魂拷问让我几乎无言以对。这是我们很多公司里都出现的场景，为做自动化而做自动化，而公司上下都并没有理解自动化测试真正的意义，也没有对自动化测试的预期。自动化的真正产出不是代码，而是更好的流程和效率。所以，自动化测试也要考虑 ROI - 投入产出比。

说到这里，又不得不聊我自己的另外一次经历。先前，被邀请到一家小型公司进行自动化梳理，老板给我提出了三大要求：

1. 自动化测试要覆盖 100% 功能测试，后续不需要再招聘功能测试人员。
2. 自动化测试要稳定，不需要太大的维护，他希望未来测试团队只有 2 到 3 名自动化人员。
3. 运行的结果可以自动报 BUG，不需要人工介入。

一般情况下，不太懂自动化的老板或者 VP，都会对自动化测试，抱有不切实际的幻想，觉得有了自动化测试可以省下很多人力成本，自动化测试能够做一切。这必然是不可能的，不要误认为

辅助性工具，再优秀的测试用例、再稳定的系统也不可能 100% 被自动化的。业务功能、复杂的逻辑、探索性的特殊操作，以及测试人员的经验，都是没有办法被自动化的。

所以，综合这两个角度来看，我们对于自动化测试的期望既不能过低，也不宜过高。那么，让我们回答之前的问题，**自动化测试的意义**到底是什么呢？谨慎来说，是把我们的自动化测试应用到项目中去，在保证质量的前提下，让项目的测试成本低于纯手工测试。

如果真的能满足这个要求，我们就可以认为自动化测试有意义了。接下来，让我们发扬咬文嚼字的风格，一起抠抠字眼，把这个要求细化一下：

首先，第一个关键词就是“**自动化测试**”。自动化测试的范畴到底有多大，在很多人看来自动化测试，就是那些可视化的 UI 层的自动化，什么 QTP、Selenium、Appium 这些东西，事实上远远不是，更不是在很多人看来，写个脚本就叫做自动化测试了。

```
#coding=utf-8
from selenium import webdriver
4 driver = webdriver.Chrome()
driver.get("https://www.baidu.com")
driver.find_element_by_id("kw").send_keys("fengluo")
driver.find_element_by_id("su").click()
driver.quit()
```

这是一段很简单的代码，打开浏览器打开百度，输入 fengluo。这当然是个自动化脚本，但是就依赖于它，我们谁都不能说我们就会做自动化了。或者再前进一步，我们能够写很长的脚本，甚至可以实现 Jenkins 的一些持续集成，就算掌握了自动化测试么？

一个好的自动化测试设计，不仅仅要能让用户方便使用，还需要符合一些基本原则。这些原则包括：可复用、易维护、定时处理、持续集成、可调试、测试结果自动通知等等。所以高效的自动化需要优秀的框架，而框架的概念，是一系列的被事先定义好的标准和规范。

在自动化测试中，我们经常提到的对测试需求的解析、脚本设计、测试执行、测试报告、维护管理等等，通过框架将它们串联并封装起来，从而使框架的终端用户，能够更方便地使用。再具体到更细节的框架我们在后边的内容里详细再介绍。

接下来是第二个关键词，“**成本**”。什么才是我们做自动化测试的成本呢？其实我们做自动化，核心的成本在于首次编写自动化脚本的时间成本，再有就是我们每一次迭代、每一次修改过程中维护自动化脚本的成本。就像前边一段说的，越是好的自动化设计，越能够进一步降低在自动化脚本编写和维护上的成本。

最后一个关键字，“**低于**”。既然是个比较级的词，那么势必要用自动化与手工测试进行一番比较。我们的基本要求是让自动化的成本低于手工，所以如果用公式来形容的话，应该是：

收益 = 手工测试成本 (自动化可覆盖部分) - 首次编写自动化脚本成本 - 自动化脚本维护成本

当收益为正的时候，则证明我们的自动化确实卓有成效。但是如果大家写过一些自动化测试脚本的话，就会发现，在瀑布式、不稳定或者是小项目流程下，这个收益很可能为负收益；这就引出了另外一个话题：既然很容易出现负收益，那么为什么要做自动化？或者换个方式，什么项目适合自动化？

放眼望去，国内外大多自动化比较成熟的公司，都有共同的特征：稳定 & 迭代。

所以像淘宝、京东这样的系统，一个版本发布以后要支持很多年，做的大多是 BUG 修改或者小的活动迭代，这样自动化的收益就会比较可观。在这种情况下，公式可以做这样的变更。

收益 = 手工测试成本（自动化可覆盖部分）迭代次数 - 首次编写自动化脚本成本 - 自动化脚本维护成本迭代次数

这样我们就会发现，只要我们在每个迭代中的维护成本小于手工回归成本，那么随着迭代次数的提高，收益就会无限增加。从这个公司我们也可以得出另外一点启示：**一个项目的早期不适合自动化**，因为需求的不断变化，业务逻辑的不断调整会使自动化的收益很低甚至为负。同时，并不是每个项目都适用于高度自动化，产品 UI 会频繁变动的，我们可能会用接口自动化来提升自动化覆盖率。

当然，这个公式只能描述大概，并不精确，通过我对很多公司的了解，我发现很多时候，我们的自动化成本中还需要增加一项：

收益 = 手工测试成本（自动化可覆盖部分）迭代次数 - 首次编写自动化脚本成本 - 自动化脚本维护成本迭代次数 - 延误成本

什么是“延误成本”？

这里边我觉得更多的是管理、支持方面带来的困难。我曾经见过很多团队的测试总监，针对一套框架不断的进行会议评审、讨论、架构调整，更有团队 leader 甚至是老板多次更换自动化测试工具，最糟糕的是还有老板要求切换语言，从 JAVA 变成 PYTHON，为什么呢？因为 PYTHON 最近火啊。一旦遇到这种情况，延误成本将变得无穷高，自动化测试的收益和效率自然不必说，只能是一场灾难。最终测试人员工作的不开心，团队领导觉得自动化测试效率低、收益差，要不

自动化还是一个很专业的领域，一个优秀的测试架构对于技术、架构和管理规划都有很高的要求。我毫不怀疑正在看到这篇文章的同学们一定有未来成为测试总监、VP 的大人物，也正像我在各种分享、在各个公司咨询时候，跟各种老板们说的最多的一句话：**无论你懂不懂技术，请尊重自动化测试技术人员和架构的判断，自动化测试并不简单，也不是可以一蹴而就的，信任你的团队，让专业的人做专业的事情。**

零零散散的说了很多，总结一下核心只有下边几点：

1. 不要为自动化而自动化，不是所有项目都需要自动化，自动化一定要保证收益才有意义
2. 自动化测试不仅仅是脚本，而是设计，通过更好的设计来降低成本提升收益
3. 国内大多公司的自动化还在路上，不要用错误的思想去认识自动化，做专业的自动化专家

你是怎么做自动化的呢？有没有什么好的方法介绍？欢迎留言大家一起讨论。

← 12 别人家的tester：一线互联网测试现状

14 回溯自动化测试工具的前世与今生 →

精选留言 5

欢迎在这里发表留言，作者筛选后公开显示

jooh123

老师 关于小白刚到测试工作，有什么建议或者学习方法吗

👍 0 回复

2020-03-05

风落几番 回复 jooh123

我觉得开始走上道路阶段，用一段时间来巩固自己的功能测试和业务精度，把测试流程体系、功能方法和特有业务上的复杂设计搞清楚，同时，尽可能的了解开发方式啊、技术架构啊等等。其实咱们在这个专栏，在这一小节之前说了很多，比如怎么高效学习、怎么问问题、怎么提升，可以想想看看，我就不赘述了，既然用了一个专栏来写，一定不是几句话可以聊明白的~~

回复

2020-03-05 14:25:47

老师，非常认同您的说法。请问您有APP自动化测试的课程吗？（我在慕课网上没找到相关课程）或者您有APP测试相关的课程推荐 不？感谢！

👍 2 回复

2019-11-13

风落几番 回复 **慕工程0182739**

APP全栈的体系课程正在计划整理中~~~请关注哇哈哈

回复

2019-11-14 16:44:21

qq_追逐梦想_8

老师分项的对自动化测试的看法很精彩！

👍 1 回复

2019-09-24

风落几番 回复 **qq_追逐梦想_8**

谢谢谢谢，欢迎大伙一块讨论哈

回复

2019-11-14 16:44:41

仲夏rww

老师，真的很想变成像您这么厉害的人呢，在被全球化浪潮冲击时能如如不动，现在好多单位能充满着不安全感，压力逐级下放，请老师一定要带带我呀！！

👍 3 回复

2019-09-19

风落几番 回复 **仲夏rww**

打铁还需自身硬嘛~加强自己才是王道！

回复

2019-09-20 16:21:02

仲夏rww

风落老师好厉害呀，我好崇拜你呀。原来做自动化要考虑这么多，不仅仅能是会写代码就行了。我现在正在学习语言，遇到了困难，就是Python的语法只能照着书敲，合上书就忘记，老师怎么能娴熟地写一手好代码，让后期的维护成本变低，请老师教教我呀。另外，公司最近经历裁员风波，每天上班有点战战兢兢的，感觉提升自己实力非常重要啊，自动化的技能学习方面，请老师给予指导！！！！感谢老师！

风落几番 回复 仲夏rww

如果说写代码的话，真的是没有捷径，只有多写多练，积攒经验才行。如果说要让后期维护成本变低，其实就是专栏里提到的，写代码过程中要有架构，有整体的思维，同时考虑怎样提取，分层，设计，让后续维护更少的针对代码，或者更少量的修改代码。

回复

2019-09-23 11:03:34

千学不如一看，千看不如一练

一手微信itit1223344