



04 | 为什么要做自动化测试？什么样的项目适合做自动化测试？

2018-07-06 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 11:32 大小 5.29M



在上一篇文章中，我为你介绍了什么是单元测试，以及如何做好单元测试，今天我来跟你聊聊什么是自动化测试，为什么要做自动化测试，以及什么样的项目适合做自动化测试。

什么是自动化测试？

不管你是刚入行的小白，还是已经在做软件测试的工作，相信你一定听说过或者接触过自动化测试。那么，自动化测试到底是什么意思呢？

顾名思义，**自动化测试是，把人对软件的测试行为转化为由机器执行测试行为的一种实践**，对于最常见的 GUI 自动化测试来讲，就是由自动化测试工具模拟之前需要人工在软件界面上的各种操作，并且自动验证其结果是否符合预期。

你是不是有点小激动？这似乎开启了用机器代替重复手工劳动的自动化时代，你可以从简单重复劳动中解放出来了。但现实呢？

自动化测试的本质是先写一段代码，然后去测试另一段代码，所以实现自动化测试用例本身属于开发工作，需要投入大量的时间和精力，并且已经开发完成的用例还必须随着被测对象的改变而不断更新，你还需要为此付出维护测试用例的成本。

当你发现自动化测试用例的维护成本高于其节省的测试成本时，自动化测试就失去了价值与意义，你也就需要在是否使用自动化测试上权衡取舍了。

为什么需要自动化测试？

为了让你更好地理解自动化测试的价值，即为什么需要自动化测试，我先来跟你聊聊自动化测试通常有哪些优势：

1. 自动化测试可以替代大量的手工机械重复性操作，测试工程师可以把更多的时间花在更全面的用例设计和新功能的测试上；
2. 自动化测试可以大幅提升回归测试的效率，非常适合敏捷开发过程；
3. 自动化测试可以更好地利用无人值守时间，去更频繁地执行测试，特别适合现在非工作时间执行测试，工作时间分析失败用例的工作模式；
4. 自动化测试可以高效实现某些手工测试无法完成或者代价巨大的测试类型，比如关键业务 7×24 小时持续运行的系统稳定性测试和高并发场景的压力测试等；
5. 自动化测试还可以保证每次测试执行的操作以及验证的一致性和可重复性，避免人为的遗漏或疏忽。

而为了避免对自动化测试的过度依赖，你还需要了解自动化测试有哪些劣势，这将帮你绕过实际工作中的“坑”。

1. 自动化测试并不能取代手工测试，它只能替代手工测试中执行频率高、机械化的重复步骤。你千万不要奢望所有的测试都自动化，否则一定会得不偿失。
2. 自动测试远比手动测试脆弱，无法应对被测系统的变化，业界一直有句玩笑话“开发手一抖，自动化测试忙一宿”，这也从侧面反映了自动化测试用例的维护成本一直居高不下的事实。

其根本原因在于自动化测试本身不具有任何“智能”，只是按部就班地执行事先定义好的测试步骤并验证测试结果。对于执行过程中出现的明显错误和意外事件，自动化测试没有任何处理能力。

3. 自动化测试用例的开发工作量远大于单次的手工测试，所以只有当开发完成的测试用例的有效执行次数大于等于 5 次时，才能收回自动化测试的成本。
4. 手工测试发现的缺陷数量通常比自动化测试要更多，并且自动化测试仅仅能发现回归测试范围的缺陷。
5. 测试的效率很大程度上依赖自动化测试用例的设计以及实现质量，不稳定的自动化测试用例实现比没有自动化更糟糕。
6. 实行自动化测试的初期，用例开发效率通常都很低，大量初期开发的用例通常会在整个自动化测试体系成熟，和测试工程师全面掌握测试工具后，需要重构。
7. 业务测试专家和自动化测试专家通常是两批人，前者懂业务不懂自动化技术，后者懂自动化技术但不懂业务，只有二者紧密合作，才能高效开展自动化测试。
8. 自动化测试开发人员必须具备一定的编程能力，这对传统的手工测试工程师会是一个挑战。

什么样的项目适合自动化测试？

看到这里，你心里可能在暗自嘀咕，“有没有搞错啊，自动化测试的劣势居然比优势还多”。那为什么还有那么多的企业级项目在实行自动化测试呢？那么，我接下来要讲的内容就是，到底什么样的项目适合自动化测试？

第一，需求稳定，不会频繁变更。

自动化测试最怕的就是需求不稳定，过高的需求变更频率会导致自动化测试用例的维护成本直线上升。刚刚开发完成并调试通过的用例可能因为界面变化，或者是业务流程变化，不得不重新开发调试。所以自动化测试更适用于需求相对稳定的软件项目。

第二，研发和维护周期长，需要频繁执行回归测试。

1. 在我看来，软件产品比软件项目更适合做自动化测试。

首先，软件产品的生命周期一般都比较长，通常会有多个版本陆续发布，每次版本发布都会有大量的回归测试需求。

同时，软件产品预留给自动化测试开发的时间也比较充裕，可以和产品一起迭代。

其次，自动化测试用例的执行比高于 1:5，即开发完成的用例至少可以被有效执行 5 次以上时，自动化测试的优势才可以被更好地体现。

2. 对于软件项目的自动化测试，就要看项目的具体情况了。

如果短期的一次性项目，就算从技术上讲自动化测试的可行性很高，但从投入产出比 (ROI) 的角度看并不建议实施自动化，因为千辛万苦开发完成的自动化用例可能执行一两次，项目就结束了。我还遇到过更夸张的情况，自动化测试用例还没开发完，项目都已经要上线了。

所以，对于这种短期的一次性项目，我觉得你应该选择手工探索式测试，以发现缺陷为第一要务。而对于一些中长期项目，我的建议是：对比较稳定的软件功能进行自动化测试，对变动较大或者需求暂时不明确的功能进行手工测试，最终目标是用 20% 的精力去覆盖 80% 的回归测试。

第三，需要在多种平台上重复运行相同测试的场景。

这样的场景其实有很多，比如：

对于 GUI 测试，同样的测试用例需要在多种不同的浏览器上执行；

对于移动端应用测试，同样的测试用例需要在多个不同的 Android 或者 iOS 版本上执行，或者是同样的测试需要在大量不同的移动终端上执行；

对于一些企业级软件，如果对于不同的客户有不同的定制版本，各个定制版本的主体功能绝大多数是一致的，可能只有个别功能有轻微差别，测试也是需要覆盖每个定制版本的所有测试；

.....

这些都是自动化测试的最佳应用场景，因为单个测试用例都需要被反复执行多次，能够使自动化测试的投资回报率最大化。

第四，某些测试项目通过手工测试无法实现，或者手工成本太高。

对于所有的性能和压力测试，很难通过手工方式实现。

比如，某一个项目要求进行一万并发用户的基准性能测试（Benchmark test），难道你真的要找一万个用户按照你的口令来操作被测软件吗？又比如，对于 7×24 小时的稳定性测试，难道你也要找一批用户没日没夜地操作被测软件吗？

这个时候，你就必须借助自动化测试技术了，用机器来模拟大量用户反复操作被测软件的场景。当然对于此类测试是不可能通过 GUI 操作来模拟大量用户行为的，你必须基于协议的自动化测试技术，这个我会在后续的性能测试章节详细叙述。

第五，被测软件的开发较为规范，能够保证系统的可测试性。

从技术上讲，如果要实现稳定的自动化测试，被测软件的开发过程就必须规范。比如，GUI 上的控件命名如果没有任何规则可寻，就会造成 GUI 自动化的控件识别与定位不稳定，从而影响自动化测试的效率。

另外，某些用例的自动化必须要求开发人员在产品中预留可测试性接口，否则后续的自动化会很难开展。

比如，有些用户登录操作，需要图片验证码，如果开发人员没有提供绕开图片验证码的路径，那么自动化测试就必须借助光学字符识别（OCR）技术来对图片验证码进行模式识别，而它的设计初衷是为了防止机器人操作，可想而知 OCR 的识别率会很低，就会直接影响用例的稳定性。

第六，测试人员已经具备一定的编程能力。

如果测试团队的成员没有任何开发编程的基础，那你想要推行自动化测试就会有比较大的阻力。这个阻力会来自于两个方面：

前期的学习成本通常会比较大，很难在短期内对实际项目产生实质性的帮助，此时如果管理层对自动化测试没有正确的预期，很可能会叫停自动化测试；

测试工程师通常会非常热衷于学习使用自动化测试技术，以至于他们的工作重点会发生错误的偏移，把大量的精力放在自动化测试技术的学习与实践上，而忽略了测试用例的设计，这将直接降低软件整体的质量。

总结

自动化测试是，把人工对软件的测试转化为由机器执行测试行为的一种实践，可以把测试工程师从机械重复的测试工作中解脱出来，将更多的精力放在新功能的测试和更全面的测试用例设计上。

然而自动化测试试一把“双刃剑”，虽然它可以从一定程度上解放测试工程师的劳动力，完成一些人工无法实现的测试，但并不适用于所有的测试场景，如果维护自动化测试的代价高过了节省的测试成本，那么在这样的项目中推进自动化测试就会得不偿失。

思考题

你在实际项目中接触过哪些自动化测试，自动化测试用例的执行比通常是多少？如果执行比过低，需要频繁更新测试用例，那你思考过你的项目是否真的适合自动化测试吗？或者说，这个项目的哪些部分更适合实施自动化测试？

欢迎你给我留言。

The image is a promotional graphic for a software testing course. It features a portrait of a man with glasses and a black t-shirt on the right side. On the left, there's a logo for '极客时间' (Geek Time) and the title '软件测试52讲' (52 Lectures on Software Testing) in large blue text, with the subtitle '从小工到专家的实战心法' (Practical Methods from Apprentice to Expert) below it. Below the title, the author's name '茹炳晟' is listed along with his title 'eBay中国研发中心 测试基础架构技术主管'. At the bottom, there's a call-to-action: '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。' (New version upgraded: Click 'Read with friends', get 10 free reads, and cash rewards for subscriptions).

© 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 03 | 什么是单元测试？如何做好单元测试？

下一篇 05 | 你知道软件开发各阶段都有哪些自动化测试技术吗？

精选留言 (54)

 写留言



Cynthia♦...

2018-07-06

15

实际项目中使用自动化的部分，接触过gui自动化测试，接口测试，性能测试。执行次数肯定是远大于5次的，毕竟开发和维护成本都要算进去，收益远超手工测试时才会考虑去做。除非是“面子工程”，用来应付某些场合。
不过还是很好奇作者的“5次”这样一个分水岭是怎么来的，是否依据经验总结得来。

展开 ▼

作者回复：5次完全完全是经验值，因为这个主要取决于自动化测试的开发成本，如果你有很好的框架，自动化用例开发的成本比较低，那么这个值就会偏小，如果有你的测试框架很低效，那么开发自动化用例的代价就很大，那么这个值自然就好大



龍蝦

2018-07-06

11

手工测试还涉及到一个人性的问题。
某些手工测试团队考核的标准就是找到的bug个数，个数越多绩效越好。而开发人员开发的代码，在一些问题上算不算bug有不同的见解。然后就开始扯皮了。

作者回复：以缺陷数量来衡量测试的绩效是不可取的，而且还会激化测试和开发之间的矛盾。



麦西尼

2018-07-06

11

在我之前参与的一个敏捷团队中，对自动化测试理解可能有些不一样，自动化测试更多的是被认为是保护已有软件功能的一张安全网，由QA和dev共同开发，和软件系统本身的功能代码一起成长，每当有新功能代码commit进主线的时候就会触发，以检测新代码是否会破坏原有功能。

作者回复：你说的是正确的，commit之后通常会自动触发静态代码扫描和单元测试，如果两周都通过后、通常会执行自动部署，然后还会自动执行smoke和e2e测试，这些都是自动化的范畴，后面的一篇文章会专门来讨论这个



Mr.Z

2018-07-06

7

现在好多公司完全把业务测试和测试开发分离开来，导致开发自动化的人不理解业务，业务用自动化工具的觉得工具不够符合业务，这样往往就是自动化成效不高，所以我一直建议测试开发要去做业务，业务要去理解怎么利用代码和工具提高效率。

作者回复：是的，你说的这个是很多公司都有的典型问题，懂自动化的不懂业务，懂业务的不懂自动化，必须要让两者能够有机的结合，否则效率很难提高，之前有提BDD其实就是为了解决这个问题，但实际落地过程中大量的mapping又引入了很多新问题



浮生凉

2018-07-06

7

我们产品迭代很快（一周一个版本）连测试用例都没办法写全，只能写写测试点，更不要提自动化了，每次刚开个头就没有然后了

作者回复：这种短期项目就不太适合自动化测试，对于这类项目的测试重点可以放在如何设计有针对性的测试用例上面，建议可以开展手工探索式测试



木主人

2018-07-11

6

自动化测试如果由自动化测试架构工程师来牵头实现，辅以业务功能的开发或测试人员构成核心团队，这样的企业级自动化测试的成本和收益应该是线性回归的：1.测试架构师负责企业级核心代码的复用设计及实现，2.项目团队内负责功能共用模块的抽取，3.两者结合建立自动化测试数据池仓库的建立，4.结合项目具体情况做自动化代码实现的二次设计。

展开 ▼

作者回复：非常棒的分享，我们以前也尝试过这种模式，效果还是不错的



Tomandy

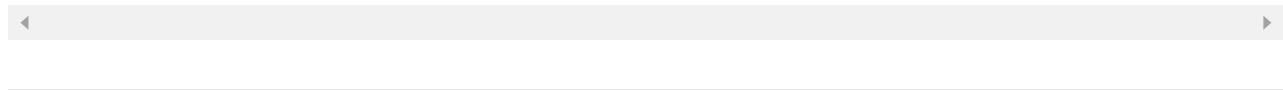
2018-07-06

5

自动化的出发点是提高效率和质量监控。盲目追求所谓的“全自动化”往往得不偿失，应根据项目实际情况做出选择。可退而求其次选择“半自动化”测试，辅助手工测试来提升效率，比如开发小工具来做资源的整合（脚本执行结果自动同步案例管理系统及缺陷系统、批量执行案例生成可视化报告、表断言检查、依赖开源框架搭建性能测试平台等）。

展开 ▼

作者回复: 说得非常对，凸



王征

2018-07-06

↑ 5

目前项目中落地的重点还是接口测试的自动化，单元测试推不动，UI自动化耗时耗力效果也不好，项目更新太快，前端页面频繁变更，不适合做UI的自动化测试

展开 ▼

作者回复: 你说的是很典型的案例，但是还是建议有最基本的GUI测试当作smoke来用，保证产品基本功能的可用性



棉花糖fami...

2018-07-11

↑ 4

老师，您好！我是一名刚毕业的学生，从事软件测试有快两个月，在公司做的是功能测试，最近看第四讲到第六讲都很懵，不知道老师能否给我些建议，如何更好的去了解消化这些知识！

展开 ▼

作者回复: 这应该不是你一个人会有这种感觉，因为第4-6讲不是基于黑盒来谈测试的，而是从软件实现的内部，即从白盒的视角来谈测试，所以需要你具有一定的代码能力，至少能够明白一门高级语言。但是你也不用太担心，因为基于代码的测试我们后面还会有专门的篇幅，那边我会以更多的实例来讲解，希望能够对你有帮助



gevin

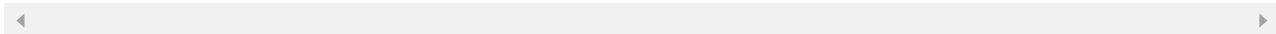
2018-07-06

↑ 3

我们这里现在是要求做api开发的后台，都要对自己的功能写测试用例，实现自动化测试，主要是出于对自己的开发功能的保护和负责，并未把自动化测试划给测试的同事来做

展开 ▼

作者回复: 这个就是目前比较流行的做法, 最理想的情况是工程效能团队可以提供更好的工具去支持开发自己做测试, 比如提供方便的测试数据准备工具, 测试执行服务等等



ThUG

2018-07-29

2

我做自动化测试有8年了, 从通信到传感器到现在的车联网。通信领域就非常适合做自动化, 无非就是造包解包, 各种场景都是可以模拟的, 可以说自动化覆盖率基本达到百分百

展开 ▼



秋荣

2018-07-09

2

我觉得比较难的部分在于如何使数据准备自动化, 尤其当需要的业务场景比较多的时候

作者回复: 是的, 非常对, 尤其是数据参数又很多的时候, 这个问题会更突出, 后面文章我会专门来讲这块



Dehom

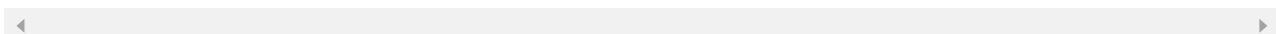
2018-07-06

2

赞同您的说法, 软件测试的目的是发现问题, 如果自动化测试维护成本高于它带来的测试效率和价值, 就失去了本身的意义。测试工程师热衷于自动化测试学习很重要的一个原因也是适应新的工作竞争吧

展开 ▼

作者回复: 所以有些时候就需要去平衡两者之间的关系



Cynthia◆...

2018-07-06

2

对了, 个人理解, CI/CD应该也算自动化测试的范畴吧, 也是应用了自动化技术, 提高手工效率的, 而且非常实用。

作者回复: CI/CD应该属于devops的范畴, CI/CD往往是自动化测试的发起者, 关于广义的我自动化测试, 在下一篇文章中我会详细来展开



口水窝

2019-02-28

1

现在想来也就是待得第一家公司做过稳定性测试, 属于一点自动化测试的范畴。后面的基本就是接口测试比较多, 且执行次数都没超过5次的, 得不偿失, 最后腰斩的很多, 且流于形式。

现在互联网迭代中周期短, 研发测试配比高, 很多小的创业公司对于测试人员可有可无的思想阶段。但是从长远看, 大的公司, 必定把质量放在第一位的, 所以, 做一个全栈...
展开▼



sylan215

2019-02-12

1

点点到位, 拳拳到肉。

非常赞同茹老师的分析, 说说我们项目的现状:

1.一直有推进自动化用例覆盖率, 但是经常改动的部分需要自动化用例频繁同步更新, 不经常改动的部分, 自动化的必要性又不大; ...

展开▼



年轻人的瞎...

2018-10-23

1

我们版本更新很快, 一般手工测试后都会进行自动化用例输出, 防止迭代之后, 开发该代码会影响到其他流程, 代码不会是硬伤→_→



欧晓鸥

2018-07-21

1

由于产品周期迭代快, 为了提高自动化投入产出比, 跟架构组, 前端组合作, 推动他们对页面重要对象加了唯一标识, 同时开发了能让手工测试人员快速加入的自动化测试框架, 与手工测试结合, 一个迭代跑5-6次左右, 能发现一些重大问题, 脚本的复用率也比较高。也为了提高收益, 准备做接口自动化, 由于公司研发团队比较新, 又推动写接口文档, 做接口测试的一些相关规范和完善接口自动化测试相关前置条件, 一切就绪, 公司业务调...

展开 ▼



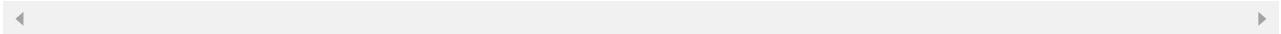
||

2018-07-12

1

测试数据的自动化在自动化测试的维护成本中有着举足轻重的作用，我认为测试数据的多样化和自动化也会给自动化测试带来了探索性和智能化的契机.....

作者回复: 非常正确，所以我们后续会专门来讲测试数据



胖虫子

2018-07-11

1

一般自动化测试都是测试部写总结时的点

展开 ▼

作者回复: 不好意思，我没看懂这句话

