



下载APP



# 开篇词 | 为什么写测试是程序员的本职工作？

2021-08-04 郑晔

[课程介绍 >](#)**讲述：郑晔**

时长 09:50 大小 9.02M



你好，我是郑晔！

看到开篇词的标题，你或许会疑惑，测试不是测试人员的本职工作吗？什么时候也成了程序员的本职工作了？

别急，让我换个问题来问问你。你说，程序员应该懂设计模式吗？大部分程序员都会说应该。而且很多人会说，这难道不是程序员的基本功吗？但你要知道，如果我们把时间往回拨，在 21 世纪初，程序员不懂设计模式才是常态，很多人会嘲笑设计模式让代码变得复杂了。



## 时代在要求我们写测试

之所以要说设计模式这个例子，主要是说**程序员的职责范围是随着时间逐步变化的**。这样的例子还有很多，迭代开发也好，开源项目也罢，这原本都不是程序员需要了解的。想当初，哪里有什么迭代开发，一个软件不开发个几年，怎么好意思出来见人。一个项目如果不是所有的代码都自己编写，怎么能有完全的掌控感，谁敢轻易使用别人开发的开源项目。而今天，哪个程序员不知道这些东西呢？

**之所以程序员的职责范围一点点在拓展，关键原因就是，软件开发正在变得越来越复杂。**而加入到程序员职责范围内的这些新东西，正是帮助程序员对抗越来越复杂的软件开发。迭代开发，让我们有机会把精力集中在最重要的功能特性上；开源项目，让我们可以减少不必要的代码编写。

测试同样如此，它可以让我们在越来越复杂的软件开发中能够稳步前行。一方面，在编写新功能时，测试可以让我们的代码正确性得到验证，让我们拥有一个稳定的模块。另一方面，测试可以帮助我们在长期的过程中不断回归，让每一步走得更稳。

程序员圈子流传着一个关于测试的段子：每个程序员在修改代码时都希望有测试，而在写代码时，都不想写测试。

希望有测试，是因为测试可以给我们带来安全感。不想写测试，一方面，很多人会觉得麻烦，另一方面，也是更重要的，团队并没有要求。为什么很多团队不要求程序员都写测试呢？这里有一个很可悲的答案，因为**大部分程序员都不会写测试**。

## 大部分程序员不会写测试

看到这个结论，可能你会说，测试有什么难的吗？不就是用 xUnit 之类的单元测试框架写代码吗？程序员每天都在写代码，写代码的事能难倒程序员，这不是开玩笑吗？

如果你把这个问题抛给一个想写测试的程序员，他会告诉你，为了学着写测试，他了解过 xUnit 的框架，甚至看过人家演示 TDD 如何去做。看别人做起来，他觉得写测试挺容易啊。

可当他有了跃跃欲试的冲动，看到了自己的代码库，所有的兴奋都烟消云散了，他还是不知道怎么写出一个测试。可能是他的代码库太复杂了，他不知道该从哪里下手；也可能是跟着别人写测试很容易，而到自己写测试的时候，他都不知道第一个测试应该从哪里开始。

有很多反对自动化测试的程序员，他会给你很多他认为自动化测试不重要的理由。但如果有机会和他深入地聊进去，你会发现，本质的答案是他不会写测试。如果你非要问他测试如何写，他只能给你一些很宏观的角度，比如从接口上去测试、按照需求去测试云云。你会发现，这些原则上正确无误的说法，其实并不能很好地指导你的工作。讨论那么多，能力不足是原罪。

## 你的代码真的是高质量的吗？

刚刚我们聊了程序员写测试是大势所趋，以及大部分程序员并不会写测试。可能这还不能够完全说服你来写测试。那么现在不妨花一分钟的时间，来仔细想想这样一个问题：你对你编写的代码有信心吗？你能拍着胸脯说这是高质量的代码吗？

等等，这不是《程序员的测试课》吗？为什么这里要说编写高质量的代码？我是走错片场了吗？相信我，你没走错。**程序员写测试就是为了编写高质量的代码。**这里所说的高质量代码分成两个部分，一方面自然是我们常规理解的：经过测试的代码，质量会更高。另一方面，要想写好测试，代码本身的质量也要高。

**对于今天的程序员来说，写测试就是程序员本职工作的一部分。**毕竟，如果你连测试都做不好，那你对自己代码的信心从何而来呢？

给你讲个就发生在我身边的故事。有一次培训，我问了一个问题，作为一个程序员，在每次代码提交之前，你对自己编写的代码很有信心的请举手。有不少程序员骄傲地举起了手。我接着问，那你的信心是从哪来的呢？一个程序员回答说，我工作这么多年了，这点自信还是有的。嗯，不错。

你在提交之前，会验证一下吗？大部分程序员的手还是高高地举着。你是验证了这次编写的代码呢？还是验证所有的代码呢？很多人一脸茫然。一个程序员说，我能保证自己的代码没问题就行了，怎么能有时间验证所有的代码呢？那你怎么保证自己写的代码没有破坏已有的代码呢？不是还有测试同学吗？我顺便问了测试同学，你们会验证系统中所有的功能吗？一个测试同学说，我们也想，但功能太多了，验证不过来。

是的，这才是大多数团队在实际开发中的真相。大多数人对于编写代码只是有一种凭空的自信，我们并不知道每次提交的代码到底有多大的影响。所以，我们常常看到在生产环境中出了问题，定位半天居然是一个简单的错误。**很多团队对于高质量代码的追求其实只是一种幻象。**

这一次，我们就来一起打破幻象，学习编写真正的高质量代码。

## 学习写测试

怎样才能学会写测试呢？**最好的办法是跟着会写测试的人一起写一段时间**，但整体行业的环境决定能提供这样机会的公司少之又少。大部分人学习测试，还是要通过阅读书籍。所以，经常有人让我推荐关于测试的书，遗憾的是，我确实没什么可以推荐的。

这些关于测试的书，要么是告诉你一些框架工具怎么用，这种东西通常看文档就能解决；要么是讲实践，比如 TDD，但还是那个问题，作者解决问题很爽，但和你有什么关系呢？归根结底，缺少一根主线，把所有这些东西连起来，让测试的知识成为一个整体。

所以，这次我准备了《程序员的测试课》，尝试把“一个程序员在日常工作中如何编写自动化测试”的相关知识梳理一遍，从实战出发，解除你对测试的一些误解，教会你一些上手可用的方法。

我把这个专栏分成了三个部分。

**基础篇**，为你讲解关于测试的基础知识。不同的是，在讲解具体的内容前，我会带你先从一个实例入手，让你看看怎么样用带测试的方式编写一段代码，告诉你一个新项目如何去测试。当我们有了对于编写测试一个直观的认识之后，再来了解具体的测试知识，就可以有更深刻的体验了。

**应用篇**，为你介绍在一个后端项目中可以怎样做测试。在这个部分，我们同样会以实战开始，主要讲解使用 Spring 框架如何做测试。之所以选择 Spring 框架，一方面，它的使用非常广泛；另一方面，它对测试提供了非常好的支持。

**扩展篇**，为你介绍 TDD 和 BDD 两项开发实践。这两项实践离很多人的实际工作是有距离的，之所以大多数人不采用这样的工作方式，思维习惯是一方面，还有一方面就是欠缺测试的基础。当我们经过这个专栏的前面部分铺垫了测试的基础之后，再来看这些实践，你会有不一样的感受。

## 写在最后

最后，还是要做一个自我介绍。我叫郑晔，一个写代码超过二十年的程序员，做过与软件开发相关的各种工作：编代码、带团队、做咨询、写开源……

我已经在《极客时间》上写了三个专栏，把自己对于软件开发方方面面的思考总结在其中。所以，在这个专栏中，你常常会看到其他三个专栏的影子：

开发代码之前要做任务分解，这是 [🔗 《10x 程序员工作法》](#) 讲过的工作原则；

代码要可测，这是 [🔗 《软件设计之美》](#) 讲过的衡量设计优劣的一个重要标准；

代码要小巧，这是 [🔗 《代码之丑》](#) 讲过的代码追求的目标；

……

所以，如果你能把这几个专栏放在一起学习，一定会功力大增。另外，这个专栏中的实战部分，也算是给老同学们的一项福利，你们呼吁的实战环节，终于在这里成真了。

不过，即便你是新同学也无妨，从头到尾学习这个专栏，你就能收获到关于自动化测试的完整认知。为了不让有些同学失望，有一点我需要提前强调一下，这个课是给程序员的测试课，而非测试人员的测试课。所以，我们这个专栏的重点是如何做好自动化测试，而不是各种测试用例的设计方法。当然，如果有测试同学想深入到自动化测试，也欢迎你的加入。

准备好和我一起编写高质量的代码了吗？欢迎你加入我的专栏，让我们一起修炼，日益精进写代码的手艺！

分享给好友，一起充电升级

👍 赞 15

🔗 提建议

© 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。

下一篇 01 | 实战：实现一个 Todo 的应用（上）

精选留言

写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。