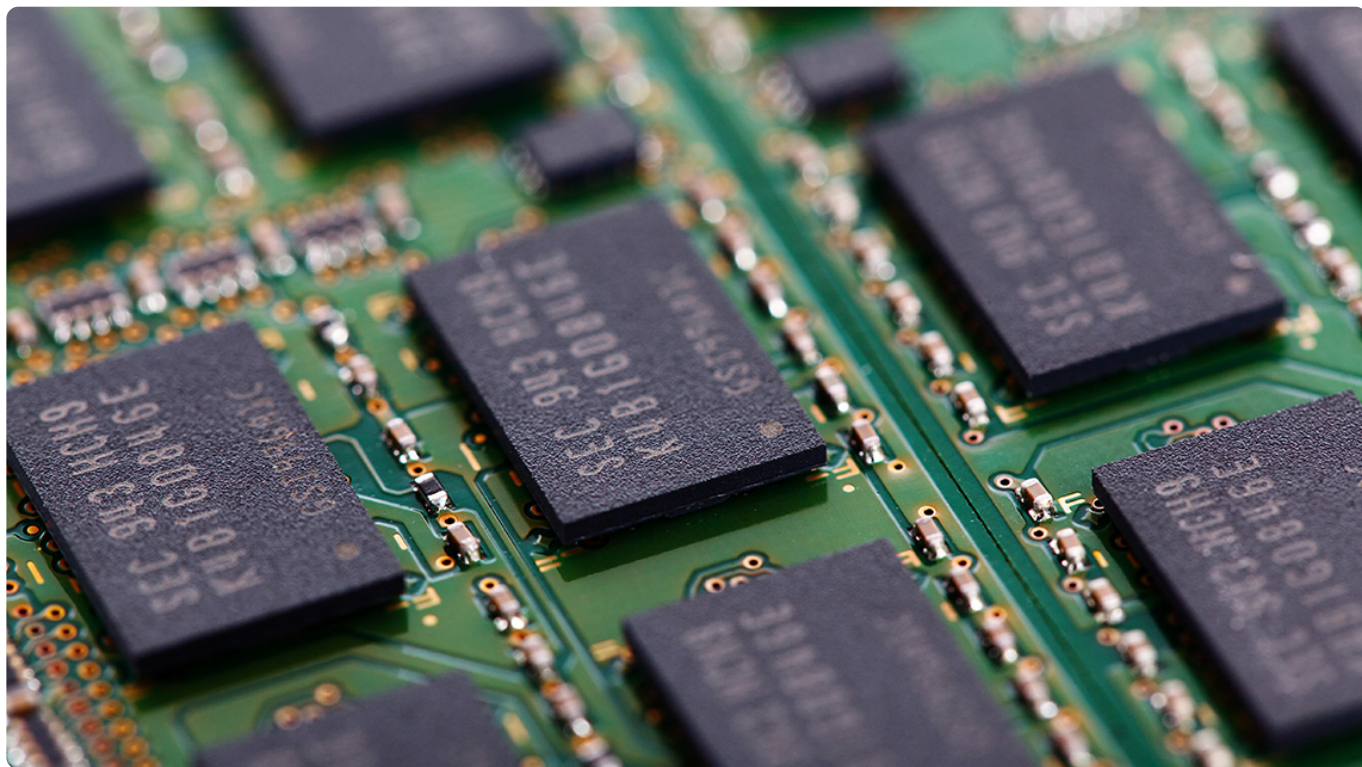


## 03 | 什么是单元测试？如何做好单元测试？

2018-07-04 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 14:27 大小 6.68M



今天我要跟你分享的主题是单元测试，如果你没有开发背景，感觉这篇文章理解起来有难度，那你可以在学完后续的“代码级测试”系列的文章后，再回过头来看一遍这篇文章，相信你会醍醐灌顶的感觉。

### 什么是单元测试？

在正式开始今天的话题之前，我先给你分享一个工厂生产电视机的例子。

工厂首先会将各种电子元器件按照图纸组装在一起构成各个功能电路板，比如供电板、音视频解码板、射频接收板等，然后再将这些电路板组装起来构成一个完整的电视机。

如果一切顺利，接通电源后，你就可以开始观看电视节目了。但是很不幸，大多数情况下组装完成的电视机根本无法开机，这时你就需要把电视机拆开，然后逐个模块排查问题。

假设你发现是供电板的供电电压不足，那你就继续逐级排查组成供电板的各个电子元件，最终你可能发现罪魁祸首是一个电容的故障。这时，为了定位到这个问题，你已经花费了大量的时间和精力。

那在后续的生产中，如何才能避免类似的问题呢？

你可能立即就会想到，为什么不在组装前，就先测试每个要用到的电子元件呢？这样你就可以先排除有问题的元器件，最大程度地防止组装完成后逐级排查问题的事情发生。

实践也证明，这的确是一个行之有效的好办法。

如果把电视机的生产、测试和软件的开发、测试进行类比，你可以发现：

电子元器件就像是软件中的单元，通常是函数或者类，对单个元器件的测试就像是软件测试中的单元测试；

组装完成的功能电路板就像是软件中的模块，对电路板的测试就像是软件中的集成测试；

电视机全部组装完成就像是软件完成了预发布版本，电视机全部组装完成后的开机测试就像是软件中的系统测试。

通过这个类比，相信你已经体会到了单元测试对于软件整体质量的重要性，那么单元测试到底是什么呢？

单元测试是指，对软件中的最小可测试单元在与程序其他部分相隔离的情况下进行检查和验证的工作，这里的最小可测试单元通常是指函数或者类。

单元测试通常由开发工程师完成，一般会伴随开发代码一起递交至代码库。单元测试属于最严格的软件测试手段，是最接近代码底层实现的验证手段，可以在软件开发的早期以最小的成本保证局部代码的质量。

另外，单元测试都是以自动化的方式执行，所以在大量回归测试的场景下更能带来高收益。

同时，你还会发现，单元测试的实施过程还可以帮助开发工程师改善代码的设计与实现，并能在单元测试代码里提供函数的使用示例，因为单元测试的具体表现形式就是对函数以各种不同输入参数组合进行调用，这些调用方法构成了函数的使用说明。

## 如何做好单元测试？

要做好单元测试，你首先必须弄清楚单元测试的对象是代码，以及代码的基本特征和产生错误的原因，然后你必须掌握单元测试的基本方法和主要技术手段，比如什么是驱动代码、桩代码和 Mock 代码等。

### 第一，代码的基本特征与产生错误的原因

开发语言多种多样，程序实现的功能更是千变万化，我可以提炼出代码的基本特征，并总结出代码缺陷的主要原因么？答案是肯定，你静下心来思考时，会发现其中是有规律可寻的。

因为无论是开发语言还是脚本语言，都会有条件分支、循环处理和函数调用等最基本的逻辑控制，如果抛开代码需要实现的具体业务逻辑，仅看代码结构的话，你会发现所有的代码都是在对数据进行分类处理，每一次条件判定都是一次分类处理，嵌套的条件判定或者循环执行，也是在做分类处理。

如果有任何一个分类遗漏，都会产生缺陷；如果有任何一个分类错误，也会产生缺陷；如果分类正确也没有遗漏，但是分类时的处理逻辑错误，也同样会产生缺陷。

可见，要做到代码功能逻辑正确，必须做到分类正确并且完备无遗漏，同时每个分类的处理逻辑必须正确。

在具体的工程实践中，开发工程师为了设计并实现逻辑功能正确的代码，通常会有如下的考虑过程：

1. 如果来实现正确的功能逻辑，会有哪几种正常的输入；
2. 是否有需要特殊处理的多种边界输入；
3. 各种潜在非法输入的可能性以及如何处理。

讲到这里，你有没有回想起我跟你分享的“等价类”。没错，这些开发工程师眼中的代码“功能点”，就是单元测试的“等价类”。

## 第二，单元测试用例详解

在实际工作中，你想做好单元测试，就必须对单元测试的用例设计有深入的理解。

通常来讲，**单元测试的用例是一个“输入数据”和“预计输出”的集合**。你需要针对确定的输入，根据逻辑功能推算出预期正确的输出，并且以执行被测试代码的方式进行验证，用一句话概括就是“在明确了代码需要实现的逻辑功能的基础上，什么输入，应该产生什么输出”。

但是，对于单元测试来讲，测试用例的“输入数据”和“预计输出”可能远比你想象得要复杂得多。

首先，让我来解释一下单元测试用例“输入数据”都有哪些种类，**如果你想当然的认为只有被测试函数的输入参数是“输入数据”的话，那就大错特错了**。这里我总结了几种“输入数据”，希望可以帮助你理解什么才是完整的单元测试“输入数据”：

被测试函数的输入参数；

被测试函数内部需要读取的全局静态变量；

被测试函数内部需要读取的成员变量；

函数内部调用子函数获得的数据；

函数内部调用子函数改写的数据；

嵌入式系统中，在中断调用时改写的数据；

...

然后，让我们再来看看“预计输出”，**如果没有明确的预计输出，那么测试本身就失去了意义**。同样地，“预计输出”绝对不是只有函数返回值这么简单，还应该包括函数执行完成后所改写的所有数据。具体来看有以下几大类：

被测试函数的返回值；

被测试函数的输出参数；

被测试函数所改写的成员变量；

被测试函数所改写的全局变量；

被测试函数中进行的文件更新；

被测试函数中进行的数据库更新；

被测试函数中进行的消息队列更新；

...

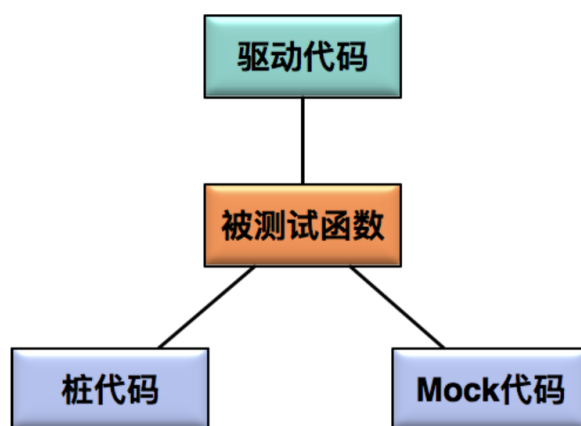
另外，对于预计输出值，你必须严格根据代码的功能逻辑来设定，而不能通过阅读代码来推算预期输出，否则就是“掩耳盗铃”了。

你不要觉得好笑，这种情况经常出现。主要原因是，开发工程师自己测试自己写的代码时会有严重的思维惯性，以至于会根据自己的代码实现来推算预计输出。

最后，我还要再提一个点，如果某些等价类或者边界值，开发工程师在开发的时候都没有考虑到，测试的时候就更不会去设计对应的测试用例了，这样也就会造成测试盲区。

### 第三，驱动代码，桩代码和 Mock 代码

驱动代码，桩代码和 Mock 代码，是单元测试中最常出现的三个名词。驱动代码是用来调用被测函数的，而桩代码和 Mock 代码是用来代替被测函数调用的真实代码的。



驱动代码，桩代码和 Mock 代码三者的逻辑关系

**驱动代码 (Driver)** 指调用被测函数的代码，在单元测试过程中，驱动模块通常包括调用被测函数前的数据准备、调用被测函数以及验证相关结果三个步骤。驱动代码的结构，通常

由单元测试的框架决定。

**桩代码 (Stub)** 是用来代替真实代码的临时代码。比如，某个函数 A 的内部实现中调用了一个尚未实现的函数 B，为了对函数 A 的逻辑进行测试，那么就需要模拟一个函数 B，这个模拟的函数 B 的实现就是所谓的桩代码。

为了帮你理解，我带你看下这个例子：假定函数 A 是被测函数，其内部调用了函数 B（具体伪代码如下）：

```
void funA ()
{
    boolean funB_returnValue;

    funB_returnValue = funB();

    if (true == funB_returnValue)
    { do Operation 1; }
    else if (false == funB_returnValue)
    { do Operation 2; }
}
```

被测函数 A 内部调用了函数 B

在单元测试阶段，由于函数 B 尚未实现，但是为了不影响对函数 A 自身实现逻辑的测试，你可以用一个假的函数 B 来代替真实的函数 B，那么这个假的函数 B 就是桩函数。

为了实现函数 A 的全路径覆盖，你需要控制不同的测试用例中函数 B 的返回值，那么桩函数 B 的伪代码就应该是这个样子的：

当执行第一个测试用例的时候，桩函数 B 应该返回 true，而当执行第二个测试用例的时候，桩函数 B 应该返回 false。

这样就覆盖了被测试函数 A 的 if-else 的两个分支。

```
boolean funB ()
{
    if (testCaseID == 'TC00001')
    {
        return true;
    }
    else if (testCaseID == 'TC00002')
    {
        return false;
    }
}
```



## 桩函数内部实现

从这个例子可以看出，**桩代码的应用首先起到了隔离和补齐的作用，使被测代码能够独立编译、链接，并独立运行。同时，桩代码还具有控制被测函数执行路径的作用。**

所以，编写桩代码通常需要遵守以下三个原则：

桩函数要具有与原函数完全相同的原形，仅仅是内部实现不同，这样测试代码才能正确链接到桩函数；

用于实现隔离和补齐的桩函数比较简单，只需保持原函数的声明，加一个空的实现，目的是通过编译链接；

实现控制功能的桩函数是应用最广泛的，要根据测试用例的需要，输出合适的数据作为被测函数的内部输入。

Mock 代码和桩代码非常类似，都是用来代替真实代码的临时代码，起到隔离和补齐的作用。但是很多人，甚至是具有多年单元测试经验的开发工程师，也很难说清这二者的区别。

在我看来，Mock 代码和桩代码的本质区别是：测试期待结果的验证（Assert and Expectation）。

对于 Mock 代码来说，我们的关注点是 Mock 方法有没有被调用，以什么样的参数被调用，被调用的次数，以及多个 Mock 函数的先后调用顺序。所以，在使用 Mock 代码的测试中，对于结果的验证（也就是 assert），通常出现在 Mock 函数中。

对于桩代码来说，我们的关注点是利用 Stub 来控制被测函数的执行路径，不会去关注 Stub 是否被调用以及怎么样被调用。所以，你在使用 Stub 的测试中，对于结果的验证（也就是 assert），通常出现在驱动代码中。

在这里，我只想让你理解两者的本质区别以确保你知识结构的完整性，如果你想深入比较，可以参考马丁·福勒（Martin Fowler）的著名文章 [《Mock 代码不是桩代码》（Mocks Aren't Stubs）](#)。

因为从实际应用的角度看，就算你不能分清 Mock 代码和桩代码，也不会影响你做好单元测试，所以我并没有从理论层面去深入比较它们的区别。

## 实际项目中如何开展单元测试？

最后我要跟你聊一下，实际软件项目中如何开展单元测试？

1. 并不是所有的代码都要进行单元测试，通常只有底层模块或者核心模块的测试中才会采用单元测试。
2. 你需要确定单元测试框架的选型，这和开发语言直接相关。比如，Java 最常用的单元测试框架是 Junit 和 TestNG；C/C++ 最常用的单元测试框架是 CppTest 和 Parasoft C/C++test；框架选型完成后，你还需要对桩代码框架和 Mock 代码框架选型，选型的主要依据是开发所采用的具体技术栈。  
通常，单元测试框架、桩代码 /Mock 代码的选型工作由开发架构师和测试架构师共同决定。
3. 为了能够衡量单元测试的代码覆盖率，通常你还需要引入计算代码覆盖率的工具。不同的语言会有不同的代码覆盖率统计工具，比如 Java 的 JaCoCo，JavaScript 的 Istanbul。在后续的文章中，我还会详细介绍代码覆盖率的内容。
4. 最后你需要把单元测试执行、代码覆盖率统计和持续集成流水线做集成，以确保每次代码递交，都会自动触发单元测试，并在单元测试执行过程中自动统计代码覆盖率，最后以“单元测试通过率”和“代码覆盖率”为标准来决定本次代码递交是否能够被接受。

如果你有开发背景，那么入门单元测试是比较容易的。但真正在项目中全面推行单元测试时，你会发现还有一些困难需要克服：

1. 紧密耦合的代码难以隔离；
2. 隔离后编译链接运行困难；
3. 代码本身的可测试性较差，通常代码的可测试性和代码规模成正比；
4. 无法通过桩代码直接模拟系统底层函数的调用；
5. 代码覆盖率越往后越难提高。

## 总结

我给你详细介绍了单元测试的概念，和你重点讨论了用例的组成，以及在实际项目中开展单元测试的方法，你需要注意以下三个问题：

1. 代码要做到功能逻辑正确，必须做到分类正确并且完备无遗漏，同时每个分类的处理逻辑必须正确；



2. 单元测试是对软件中的最小可测试单元在与软件其他部分相隔离的情况下进行的代码级测试；
3. 桩代码起到了隔离和补齐的作用，使被测代码能够独立编译、链接，并运行。

## 思考题

你所在的公司有做单元测试吗？实施单元测试过程中遇到过哪些问题，你是如何解决的？

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 02 | 如何设计一个“好的”测试用例？

下一篇 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？

## 精选留言 (50)

 写留言



小志

2018-07-04

 34

我待过一些中国的互联网公司，也问过一些创业公司，进行单测的不多。我也一直在想没有单测，但是整体的质量也还能“说的过去”的原因是什么。首先没有单测的主要原因还是和中国互联网的现状有关，中国的互联网本质是商业公司，在中国激烈的竞争环境下，业务的快速发展导致需求的快速上线是一个常态。这也导致了无论是产品还是开发经理，都是以支撑业务为kpi，其次才是质量。所以单测是一件不能直接体现到kpi上的隐形需...  
展开 ∨



Cynthia◆...

2018-07-04

👍 11

之前的一家公司曾经比较重视单元测试，当时具体单元测试代码是开发写的，对于测试部门的我而言，只是在CI这块，负责跑出各项目的单元测试结果后汇总成报告查看。具体单元测试的质量是由开发进行把控和审核。

再后来组织有些变动，不再重视单元测试，便流于形式，开发可能就写个假的代码保证跑出的报告好看，实际的单元测试本身则被忽视了.....

展开 ∨



BADBOY

2018-07-04

👍 6

老师，我的学习方向是Python接口自动化，JMeTer，LR，是不是在学做性能之前就得先去学Java呢，在这学习方向中有什么建议么！谢谢



黑米

2018-11-05

👍 5

我们开发了个单元测试框架，测试人员写用例只要修改yaml配置文件即可完成单元测试，比以前方便多了。



Jump

2018-08-17

👍 5

我现在的公司是比较重视单元测试的，也有专门也单测的人，我就是其中一个，我们单测内容主要分两部分，一是基于mock的真正意义上的单元测试，这部分主要验证业务逻辑，二是与数据库的集成，应该算是集成测试，主要验证数据方面的一致性。另外，看评论有一位说在推行c#的单元测试，我们这边就是用的c#，框架之前用的Nunit,由于数据隔离性不是特别好，后来又换成了Xunit框架，Mock框架是moq，或许可以和那位哥们讨论一...

展开 ∨



Geek\_84a77...

2018-07-04

5

不太理解老师说的输入数据那部分  
只知道被测函数的参数进行设计  
不知道如何针对函数的成员变量等进行设计用例?

展开

作者回复: 能问这个问题, 说明你已经很好地理解了文章的关键内容。这里的成员变量指的是类的成员变量, 逻辑上你也可以把它想象成是全局变量。因为函数内部会去读取类的成员变量, 然后根据类的成员变量来决定后续逻辑等。



海罗沃德

2018-07-05

4

在Accenture时候单元测试行覆盖率要达到99%, 分支覆盖率要达到90%, 仅一些exception分支可以不用覆盖, 并且每个测试用例前面要注释好这个case测的是什么方法, 输入什么输出什么, 预计结果是什么等以便code review时可以快速的知道这段代码是做什么的, 甚至一些大功能还要带上use case的id方便追溯原始需求, 在测数据持久化层测试要通过内存数据库把CRUD流程都测出来

展开

作者回复: 埃森哲的所有项目都有这么高的单元测试需求吗? 我的理解是一些和人的生命安全息息相关的软件才会有几近苛刻的代码覆盖率要求, 比如航空航天, 汽车电子, 轨道交通, 部分医疗器械软件等, 如果所有项目都这样做成本还是很高的



sylan215

2018-08-16

3

十分赞同茹老师的这个观点「并不是所有的代码都要进行单元测试, 通常只有底层模块或者核心模块的测试中才会采用单元测试。」, 这一定是首要前提, 不然在落实的时候会发现, 被测函数可能会完全被 Mock 代码取代了。

如果要做单元测试, 那么对开发代码的要求也会更高, 至少开发在代码分层上一定要做...

展开



口水窝

2019-02-26

2

同在小的创业公司，根本不用问，没做单元测试，最近在学茹老师的课程，java基础知识，有研发看到，竟然说我一个测试干嘛去学开发的东西，我无语，我只能按照自己内心的想法去学习，知道自己想要什么，然后去努力，坚持打卡，加油！

---



谭鹏

2019-02-27

👍 1

公司的测试 都是靠开发人员自测和测试人员测试，测试方法也是手动点击的这种笨方法,想写打哪元测试，同事们都说 代价太大，最后也没执行起来。感觉测试集成也是一套系统，光靠单个方法和技术点没有完整的系统 感觉就像是增加了工作量

---



Xhavit

2018-12-19

👍 1

倒是定义一下mock代码啊？都不定义然后就对比，一脸懵逼。。。。

展开 ▾

---



zhangbita...

2018-08-31

👍 1

我在开发过程中会写单测，但是耦合性一直是个大问题，有时候想到很多测试用例，但是模拟起来很麻烦就放弃了

---



coco

2018-07-12

👍 1

测试review ut也许不能提出很多意见，而且要开发来讲解一下，但是有两个好处1.防止假测试用例，也就是不做验证的用例，2.在进行上层用例的设计时可以知道哪些已经在ut层覆盖了，避免重复劳动

展开 ▾

---



卫宣安

2018-07-04

👍 1

今年刚在团队推行单元测试，阅读过《单元测试的艺术2》，觉得非常受益，也强烈推荐其他同学阅读。我认为单元测试不仅仅是为了测试，也能让你写出结构更好，质量更佳的逻辑代码。在推行的这几个月，也只能以新代码进行试水，遗留代码完全没有勇气进行。而且目前团队成员在接受程度上还远没有达到得心应手，也比较容易出现抵触情绪，我也正在思考如何才能更有效的推广。...

展开 ∨



产品助理

2018-07-04

👍 1

项目中推行单元测试中，您提及的问题如何解决，后面会有介绍吗？

如单元函数中大部分都是对数据的CURD操作，如何获取有效数据，又如何防止脏数据。都很让人头痛。

...

展开 ∨



永不放弃

2018-07-04

👍 1

老师，后面会有实战部分吗？

展开 ∨

作者回复: 专栏后期会有专门的篇幅讲代码级测试，那里会提供更多实际的例子



Tomandy

2018-07-04

👍 1

公司有套CI CD的流程，确实用到了文中提到的单元测试和代码覆盖率。

展开 ∨



辰九

2019-06-03

👍

单元测试的有效性如何保证呢？覆盖率上去了是否真的说明代码质量就高了？

展开 ∨



奕

2019-05-09

👍

自动化测试集应该是一把可信的、灵活的尺子。所以测试集不宜过大，应能支持在几个小时内给出稳定可信结果。测试集的大小应考虑以下几个方面：以时间窗口为首要敏感因素，然后考虑覆盖功能的重要程度，测试执行的稳定性。

展开 ∨

作者回复: 很棒的总结



**Daisy**

2019-04-10



我遇到过这样的问题，我做单元测试是独立新建了一个测试工程，使用testng框架，无法使用mock，只能在开发代码的原工程写test才可以使用mock