

为什么要讨论这些概念？

不止一次听到同学疑惑地问我：“我们为什么要纠结于这些基本概念，干嘛不赶紧上测试方法，赶紧教我怎么用工具，扯这些没用的概念干嘛呢？”下面，让我们来看下面的这段对话。

小测：老师，软件测试、软件缺陷、测试用例、软件质量这些概念跟我有啥关系呀？干嘛这么纠结于这些没用的概念呢？难道不应该开门见山，赶紧谈谈怎么设计测试，怎么测一个复杂的软件产品，特别是怎么使用五花八门的测试工具，这些才是我感兴趣的呀。

老师：我可以理解你迫切想开始做测试的心情，不过，凡事都需要从扎实的基本功开始。举个例子吧，在下水游泳之前，我们需要先了解水是有浮力的，人在水中是可以浮起来的，从而克服对水的畏惧感；我们还需要了解只要在水中屏住呼吸，水是不会进入鼻腔而造成呛水的，这样我们就知道只要不呼吸，就可以在水中通过手脚协调划水往前游。至于蛙泳、自由泳等各种游泳方法，区别只是在于换气的时机、游泳的姿势等不同而已。

小测（点了点头表示认可，随即又问）：嗯，我了解了。可是，我还是觉得什么是软件测试，什么是软件缺陷，这些概念与如何做测试看起来没什么关系，我 Get 不到点啊。

老师：其实这些概念与测试方法是有关联的。就 IEEE 对软件测试所下的定义来说，这个定义最重点的贡献在于它让测试工程师与开发工程师的工作目标始终保持一致，那就是**确保软件产品符合用户需求**。因此，初始的测试就是依据用户需求（通常是需求规格说明书）来设计并执行测试，或者模拟用户使用软件产品的方式来操作被测软件以达到测试软件产品的目的。

小测（插嘴道）：那既然是这样，还需要测试方法干什么呢？我们就看用户需求去做测试不就足够了？

老师（无奈地摇了摇头）：但是时间和成本都不允许我们这样做呀。我们在谈软件测试定义的时候提到过，项目团队在开发和测试中所依据的需求**不是原始用户需求**，只是**软件需求**而已，用户需求经常就是一个坑啊。所以，我们必须在与用户（或客户）商量好的时间内尽快完成开发和测试，还得控制好成本，这就要求我们使用更好的测试方法来设计测试用例，才能满足效率的要求。拖的时间越长，用户改变需求、甚至增加需求（但不增加投资）的概率越高，对项目团队越不利啊。

小测（诧异地啊了一声）：啊，这么多坑。那软件缺陷还需要正式定义吗？是不是缺陷，难道还看不出来吗？缺陷的定义与如何做测试哪有关联呢？

老师：是不是缺陷关键在于需求是什么，以及犯错的究竟是**使用软件的人**，还是**被人使用的软件**。

小测（挠了挠头）：老师，你在说绕口令吗？

老师（笑起来）：哈哈，如果是使用软件的人，也就是用户没有发出正确的指令，换句话说，用户没有正确地操作软件，此时软件无法判断用户的真实需求，而是按照该提供的功能加以执行，此时就不能说是软件缺陷。例如，汽车驾驶员的本意是停车，结果错将油门当做刹车踏板踩下去，出错的是人不是车。软件缺陷主要是指被人使用的软件在用户发出正确指令后没有正确地提供应有功能的情况。无人驾驶的那个视频中，就存在这种情况，你有空可以去看看软件缺陷那节的案例讨论，看你能找出哪些软件缺陷，能找出哪些是人犯错而不应视为是软件缺陷的情况。

小测（点点头）：好的。

老师（继续说下去）：Ron Patten 给出的这个软件缺陷的定义给我们指出了判断是否为缺陷的五个明确的思路，除了遵循用户需求这个思路，其他四个思路都是告诉我们，该如何根据需求规格说明书有针对性地展开测试。而且，这个定义也是告诉我们，有一些软件缺陷是测试工程师没法发现的，在需求不充分、程序员自作主张、用户私下找开发改需求等情况下，总有软件缺陷是会从测试工程师背后溜到用户那里去的，不能是只要用户报错，就让测试工程师背锅。

小测：额，可怜的测试，希望产品经理能看到这段话，不要动不动就说为啥你做测试的没测出来……那，软件质量那个定义呢？感觉离我一个做开发的很遥远啊。我只要保证做设计的时候注意一点就可以了，质量什么的不是测试、QA 之类的人该负责的吗？跟我不相干啊。

老师（眨了眨眼）：你说你的代码质量好，有证据吗？

小测（一脸迷惑）：证据？每次课程设计，开发任务都是我包了，我们小组我就是“第一大腿”，框架啥的，我都很熟悉，我的开发水平有目共睹，这不是证据？

老师（微笑地望着小测）：但为什么每次你们小组作业都拿不到 A 呢？

小测（眉头紧锁）：别提了，每次还不都是因为小查？他是我的好兄弟，我不能不带上他，但他总是拖后腿，他负责的模块总是一堆问题，最后经常害得我改来改去改不好，有时候就

只能自己重新写，可时间又来不及了，所以最后的总体表现总是差火候。我让他提高编程水平，他说不知道哪里做的不好，没法提高。我有什么办法。

老师（微笑）：对呀，软件质量就是要确保你能提供证据，告诉别人，你的代码质量就是好；软件质量同时也能为团队所有人提供标准，让大家必须保证达到这个统一的标准，而且，哪方面做的不好也一目了然，这样就可以有目的去提高了呀。所谓**测试只能检验软件质量，预防建立质量**就是这个道理。

小测（兴奋地跳起来）：好，好，好，到底该怎么做呢？赶快告诉我吧。

老师（笑起来）：心急吃不了热豆腐。我们在白盒测试单元，以及课程最后两周会就这个问题做进一步讨论。下周我们将开始讨论黑盒测试方法，提供更有效的手段来展开测试。你有空可以先尝试只利用软件缺陷的定义来设计测试用例，正好与下周讨论的方法的效果做个对比。

小测（陷入沉思）：好的，我先试试，谢谢老师。

老师（微笑）：不客气。以后找时间再聊。