我们作为软件开发人员,为什么要编写单元测试?

**定义：单元测试是用来对一个模块、一个函数或者一个类来进行正确性检验的测试工作。**

软件测试是为了保证项目质量，单元测试可以快速执行测试回归测试，做好单元测试可以大大提升测试效率，项目开发真正达到敏捷效果。

单元测试的作用：

****（1）让我们对自己的代码有信心****

修改了代码后单测依然通过的，起码说明我们的修改没有破坏程序的正确性。这从主观上能增加我们对代码的信心。虽然单元测试通过了并不意味着程序就没有bug了，但我们也要了解到这可能不是单元测试的问题。单元测试顾名思义是测试一个”单元”，这个”单元”一般是类或方法，而不是整个系统。对整个系统的测试那是集成测试，功能测试的职责。单元测试追求的是快速反馈，频繁执行。集成测试虽然测“全局”，但成本较高，所以执行频率较少。两者使用场景不同，目的不同。

（****2）为代码重构保驾护航****

看到代码很差劲，想重构，但又担心重构之后出问题，怎么办呢？如果有单元测试情况就不一样了，重构完代码，跑一遍单元测试，如果单元测试都通过，基本上可以保证我们的重构没有破坏原来代码逻辑的正确性。不过前提是之前的写的单元测试质量很好，覆盖率很高。当然这仅限于小范围的重构，比如重构一个类或者函数的实现，但对于大刀阔斧的重构（比如单体重构成微服务，面向库表模式重构成DDD），就不适用，那个时候要重写单元测试了。

****（3）通过单元测试快速熟悉代码****

单元测试不仅起到了测试的作用，还是一种很好的“文档”，通过单元测试，我们不需要深入的阅读代码，便能知道这段代码做什么工作，有哪些特殊情况需要考虑，包含哪些业务。（摘自CSDN）

****（4）单元测试是bug侦测器****

良好的单元测试是强大的bug侦测器，能够大大缩减查找bug所需要的时间。因为修改了代码引入了新的bug，执行单元测试时会立刻出现错误。由于修改前的代码是可以正确运行的，那么错误就一定出在最近修改的这部分代码中，而刚刚修改的代码我们记忆犹新，并且分量也很小，所以很轻松就能找到错误。以前需要2小时才能找到的错误，现在只需要5分钟。

****（5）单元测试可以提高代码质量****

这可能会让一些人困惑，测试与代码质量有什么关系呢？单元测试是你开发的功能的第一个用户。只有在使用代码时你才会发现代码的缺点。如果你的代码难以测试，那么它通常也难以使用。单元测试可以使你发现程序结构上问题，并在发布前优化。（摘自《简书》 作者：柴诗雨）

参考Google的开发架构，Google有测试开发工程师，但是其测试开发工程师，都是原来开发年限较长的开发转岗位，他们对自己曾经的开发领域是非常熟悉的。所以他们写的单元测试用例是非常细致，考虑非常周全的，同样他们也可以进行对应代码的修改与开发，这有点结对编程的意思了。Google 的开发一直都有写单元测试的传统与习惯，中国的开发极少有这样的思想。

所以真正全面做单元测试，必须调整开发结构，说服开发，自顶向下执行，养成单元测试习惯，逐步提高测试效果，保证项目质量。

（查了半天终于明白为什么要做单元测试了！！）