单元测试介绍与推广意义

单元测试由一组独立的测试构成，每个测试针对软件中的一个单独的程序单元。单元测试并非检查程序单元之间是否能够合作良好，而是检查单个程序单元行为是否正确。

在单元测试时，测试人员根据详细设计说明书和源程序清单，了解到该模块的I/O条件和模块的逻辑结构，主要采用白盒测试的测试用例，辅之以黑盒测试的测试用例，使之对任何合理和不合理的输入都要能鉴别和响应。这就要求对程序所有的局部和全局的数据结构、外部接口和程序代码的关键部分进行桌面检查和代码审查。

但是,对于单元测试,人们往往存在很多的误解:

1）浪费的时间太多

一旦编码完成，缺乏软件工程实践经验的开发人员就会迫不及待地进行软件集成工作，这样就能看到实际系统开始启动工作，在这种开发步骤中，真正意义上的进步被表面上的进步所取代。系统能进行正常工作的可能性很小，更多的情况是充满了各式各样的Bug。这些Bug包含在独立的单元里，其本身也许是琐碎、微不足道的，但在软件集成为一个系统时会增加额外的工期和费用。

其实进行完整的单元测试和编写代码所花费的精力大致上是相同的，一旦完成了单元测试，在确保手头拥有稳定可靠部件的情况下，再进行高效的软件集成才是真正意义上的进步。

程序的可靠性对软件产品的质量有很大的影响，在大型软件公司，每写一行程序，都可能要测试很多遍。由此可见大型软件公司对测试的重视程度。

2）软件开发人员不应参与单元测试

单元测试常常和编码同步进行，每完成一个模块就应进行单元测试。在对每个模块进行单元测试时，不能忽略和其他模块的关系，为模拟这一关系，需要辅助模块，因此若单独的测试人员进行单元测试，往往工作量大，周期长，耗费巨大，其结果事倍功半。软件的开发者总是应当负责程序的单个单元的测试，保证每个单元能够完成设计的功能，其实在很多情况下，开发者也应进行集成测试。

3）我是很棒的程序员，不需要进行单元测试

如果我们真正擅长编程并且有绝招，就应当不会有错误，但这只是一个神话。编码不是可以一次性通过的，必须经过各种各样的测试，单元测试只是其中一种。缺乏测试的程序代码可能包含许多Bug，程序员在没有测试保护的情况下修改Bug，会引发更多的Bug，忙于除虫，于是更没有时间测试。如此循环往往会导致项目的崩溃。为避免产生恶性循环，代码必须有一张安全网来保护，随时进行的单元测试就是这张安全网。

4）不管怎样，集成测试将会抓住所有的Bug

集成测试的目标是把通过单元测试的模块拿来，构造一个在设计中所描述的程序结构，通过测试发现和接口有关的问题。我们在测试工作开展的过程中，发现并提交进行合格性测试的软件，在测试过程中有很多Bug，有些严重问题，甚至导致死机，以至于不能再测试其他功能，进行错误修改，回归测试时又发现其他新的问题，使得测试工作很难开展下去。

5）单元测试效率不高

在实际工作中，开发人员不想进行单元测试，认为没有必要且效率不高，其实错误发生和被发现之间的时间与发现和改正该错误的成本是指数关系，频繁的单元测试能使开发人员排错的范围缩得很小，大大节约排错所需的时间，同时错误尽可能早地被发现和消灭会减少由于错误而引起的连锁反应。

在某一功能点上进行准备测试、执行测试和修改缺陷的时间，单元测试的效率大约是集成测试的两倍、系统测试的三倍。

通过对这些误解的分析，可以对单元测试有进一步的了解，其实作为软件系统的最小组成单位，单元测试具有以下属性：

它主要是由一个程序员完成的。

它有一个详细的设计说明，包括输入定义、输出定义和加工说明。

它是一个可识别的看得见的程序组成部分，并容易被组合成程序。

能被单独地汇编和测试。

它的规模比较小，逻辑比较简单。

因此推广单元测试具有以下意义：

1）单元测试集中注意力于程序的基本组成部分，首先保证每个单元测试通过，才能使下一步把单元组装成部件并测试其正确性具有基础。单元是整个软件的构成基础，像硬件系统中的零部件一样，只有保证零部件的质量，这个设备的质量才有基础，单元的质量也是整个软件质量的基础。因此，单元测试的效果会直接影响软件的后期测试，最终在很大程度上影响到产品的质量。

2）单元测试可以平行开展，这样可以使多人同时测试多个单元，提高了测试的效率。

3）单元规模较小，复杂性较低，因而发现错误后容易隔离和定位，有利于调试工作。

4）单元的规模和复杂性特点，使单元测试中可以使用包括白盒测试的覆盖分析在内的许多测试技术，能够进行比较充分细致的测试，是整个程序测试满足语句覆盖和分支覆盖要求的基础。

5）单元测试的测试效果是最显而易见的。做好单元测试，不仅后期的系统集成联调或集成测试和系统测试会很顺利，节约很多时间；而且在单元测试过程中能发现一些很深层次的问题，同时还会发现一些很容易发现而在集成测试和系统测试很难发现的问题；更重要的是单元测试不仅仅是证明这些代码做了什么，是如何做的，而且证明是否做了它该做的事情而没有做不该做的事情。

6）单元测试的好与坏不仅直接关系到测试成本（因为如果单元测试中易发现的问题拖到后期测试发现，那么其成本将成倍数上升），而且也会直接影响到产品质量，因为可能就是由于代码中的某一个小错误就导致了整个产品的质量降低一个指标，或者导致更严重的后果。

事实上，单元测试是一种验证行为—— 测试和验证程序中每一项功能的正确性，为以后的开发提供支持；单元测试是一种设计行为—— 编写单元测试将使我们从调用者观察、思考，特别是要先考虑测试，这样就可把程序设计成易于调用和可测试的，并努力降低软件中的耦合，还可以使编码人员在编码时产生预测试，将程序的缺陷降低到最小；单元测试是一种编写文档的行为—— 是展示函数或类如何使用的最佳文档；单元测试具有回归性—— 自动化的单元测试有助于进行回归测试。