**Visual Studio 中的单元测试 UNIT TEST**

TDD(Test-Driven Development) 测试驱动开发是敏捷开发中的一项核心实践和技术，也是一种设计方法论。TDD的原理是在开发功能代码之前，先编写单元测试用例代码，测试代码确定需要编写什么产品代码。单元测试是最基本的测试步骤。位于整个产品开发流程V模型的最底部。

大致如图，在各种开发流程中RA&PSD完成后，无需底层基础，UAT方案即可完成，下面的步骤也一样。UT做为底层和CODING同样重要的步骤，是每个CODER必须掌握的基础。

RA( Requirement Analysis) &

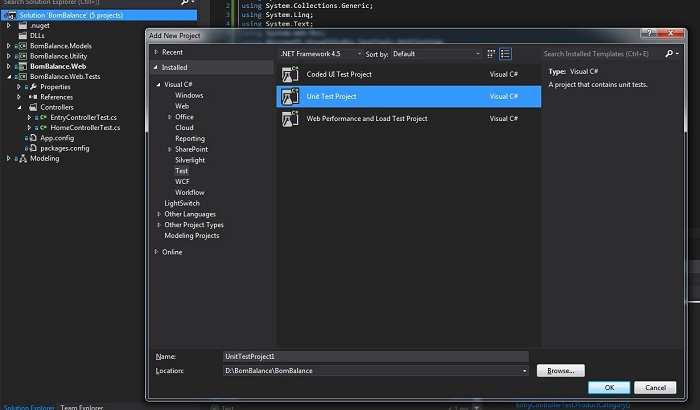
PSD ( Productive Specification Design) ----FSD(Functional Specification Design)--- Construction(Coding)

UAT( User Acceptance Testing)--- SIT(System Integration Testing)---UT(Unit Tesing)

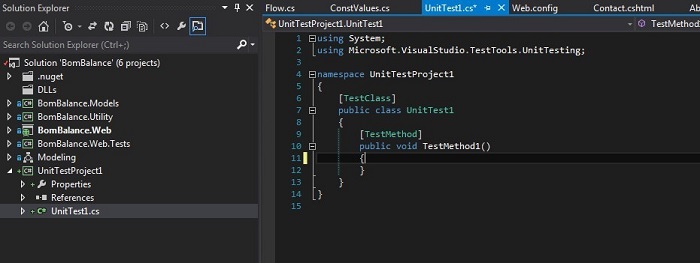
在小量代码编写时，往往可以通过新建控制台项目(Console Application)，新建网站项目(Web Form)等，在其中敲入测试代码来进行测试。一旦代码量及逻辑关系变得复杂，这种测试将难以展开，并且不具备可维护性。而且很多功能性的代码，不能完全脱离整个项目来实现。比如在某WEB项目中有一个类库，写了一些MODEL层的方法，为了验证方法的正确性，若要测试该方法，可能不得不引用项目中大量的模型，接口，以及库函数。等到编译可以通过了，却发现你必须登录才能调用该方法或者您所在的目录(Directory)根本不具备访问权限，于是又不得不去修改Authorization配置或者添加登录等方法。于是一层层下来，你的测试几乎涉及的项目的各个角落，而该测试方案又不具备高度的重用性。  
下面我们以VS2012为例，来看一下如何在Visual Studio中进行单元测试。

1.首先，右键点击解决方案（Solution)弹出右键菜单(Context)。

选择添加(Add) - 新项目(New Project)， 在给出的模版中，选择 Visual C# - Test -Unit Test Project 如图。

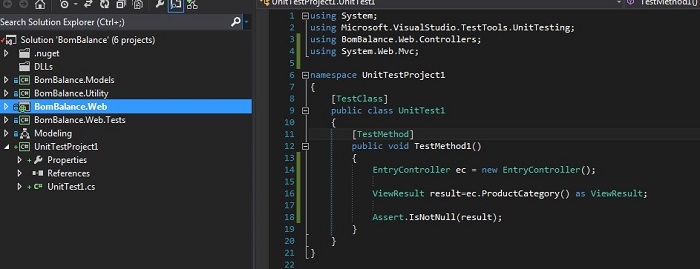


2.得到模版如图。

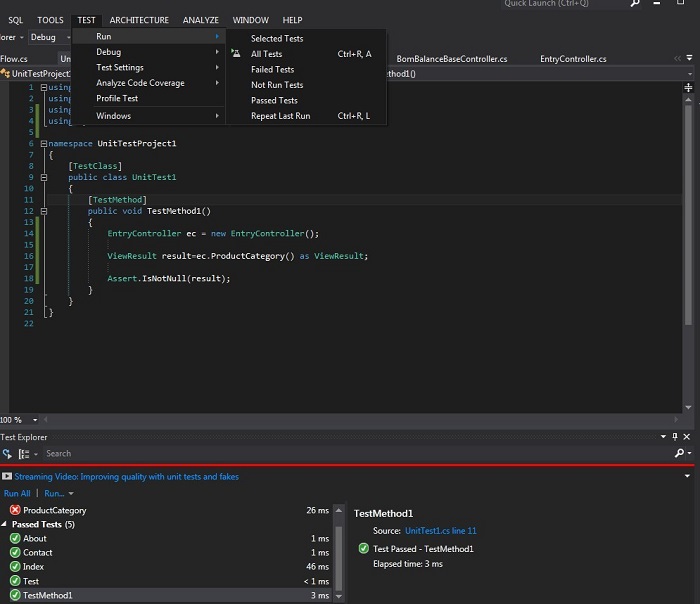


3.在测试方法中(此处为默认的TestMethod1，一般修改为 需要测试的方法名+Test ）添加自己需要测试的代码。

不过在此之前，首先要引用需要测试的项目以及需要用到的类库。 此例中，我引用了微软的MVC4.0框架，以及需要测试的项目中的Controller。在此我测试该Controller的ProductCategories()方法。该方法返回一个ActionResult。测试代码中有3行，第一行用来实例化controller，第二行调用controller的方法。如果仅以该2行进行测试通过，则说明代码无语法问题，可以和其他各部分配合工作。第三行用了一个断言来检测方法返回的result是否为空。此处我断言它非空，如果为空，测试将会失败报错。



4。测试代码写好后， 即可点击上方菜单Test-Run- AllTests等，来进行测试。



测试完毕后。下方会产生结果列表。红色为未通过的TestCase。若想对其进行DEBUG，可右击红色的TestCase，选择Debug selected Tests。修改后，也可右击想要重新测试的TestCase，选择Run Selected Tests.