

- **Каждый проект должен иметь свой проект с тестами**
 <PROJECT_NAME>.Core -> <PROJECT_NAME>.Core.Tests
 <PROJECT_NAME>.BI -> <PROJECT_NAME>.BI.Tests
 <PROJECT_NAME>.Web -> <PROJECT_NAME>.Web.Tests.

- **Имена для классов**

Например класс ProblemResolver должен быть проименован как ProblemResolverTests. Каждый тестирующий класс должен тестировать только одну сущность.

- Для имен методов при тестировании используем следующий шаблон **ShouldExpectedBehaviorWhenStateUnderTest**. When может быть пропущено

Например:

- ❖ ShouldThrowExceptionWhenAgeLessThan18
- ❖ ShouldFailToWithdrawMoneyForInvalidAccount
- ❖ ShouldFailToAdmitIfMandatoryFieldsAreMissing

- **Метод внутри должен быть разбит на три составляющих *arrange, act, assert* - AAA принцип.**

```
class CalculatorTests
{
    public void ShouldReturn7For2Plus5()
    {
        // arrange
        var calc = new Calculator();

        // act
        var res = calc.Sum(2, 5);

        // assert
        Assert.AreEqual(7, res);
    }
}
```

- **Каждый тест должен проверять только одну вещь.**

Если процесс слишком сложен (например, покупка в интернет магазине), разделите его на несколько частей и протестируйте их отдельно.

Если не придерживаться этого правила, тесты станут нечитаемыми, и вскоре окажется очень сложно их поддерживать.

- **Fakes: stubs & mocks**

Часто эти понятия путают. Разница в том, что стаб ничего не проверяет, а лишь имитирует заданное состояние. А мок – это объект, у которого есть ожидания. Например, что данный метод класса должен быть вызван определенное число раз. Иными словами, ваш тест никогда не сломается из-за «стаба», а вот из-за мока может.

С технической точки зрения это значит, что используя стабы в Assert мы проверяем состояние тестируемого класса или результат выполненного метода. При использовании мока мы проверяем, соответствуют ли ожидания мока поведению тестируемого класса.

- **Тестироваться должен ТОЛЬКО текущий класс.**

Если начинается тестироваться методы, которые данный класс получает из зависимостей, то тест перестает быть unit-тестом и становится интеграционным. Зависимости должны быть подменены на fake-реализацию.