Unit testing

Чрез unit тестовете тестваме само отделни компоненти от програмата (отделни методи). Чрез unit тестовете не се тества цялата функционалност на един клас или на самата програма (end-to-end testing)

Unit тестове – пишат се от самия програмист (не от QA)

Обикновено се пишат в отделни проекти

За да се пишат unit тестове е нужен framework. Във Visual Studio има вграден framework за unit тестове – Visual Studio Team Test

Какво трябва да тестваме: трябва да се тестват public класовете, public методите, понякога може да тестваме и setter-ите

Unit tests:

* Dramatically decrease the number of defects in the code
* Are good documentation
* Allow easier refactoring

Unit test-ването, конкретно в C#, се извършва чрез assertion-и

Добре е един тест метод да тества едно нещо, а не няколко неща, т.е. да има 5, 10 или повече Assert-а, който тестват различни по характер неща. Един поблем, който може да възникне в такъв случай е, че няма да можем да намерим правилно име на тест метода, и друг проблем е, че ако първия поред Assert не се изпълни, до другите Assert-и след него въобще няма да се стигне

Всеки един testing framework работи с атрибути, така че атрибутите трябва да се слагат над съответния клас, метод и др.

Голяма част от времето отнето за даден софтуер (около 50%) отива за тестване

Example of VSTT assertion:

**Assert.AreEqual(expectedValue, actualValue, "Error message.");**

**Examples:**

**AreEqual(expected\_value, actual\_value [,message])**

* Comparing values for equality

**AreSame(expected\_object, actual\_object [,message])**

* Comparing objects (by reference)

**IsNull(object [,message])**

**IsNotNull(object [,message])**

* Checking for null value

Checking conditions:

IsTrue(condition)

IsFalse(condition)

The correct way to write unit tests is to use “the 3A pattern”

* Arrange all necessary preconditions and inputs
* Act on the object or method under test
* Assert that the expected results have occurred

Нормално е след променяне на дадена функционалност на нашия код, някой тест, който сме написали преди това, да се счупи. В този случай не означава, че програмата ни е грешна, а че самия unit тест ни е грешен, и трябва или да променим или да изтрием конкретния unit тест

Когато даден Assert не е достатъчно показателен за това какво точно прави, е добре да подадем достатъчно обяснително съобщение като параметър на този Assert

Reflection – изследване на кода, който пишем чрез друг код