

Best Practices

Inleiding

Ontwikkelaars hebben vaak een liefde-haat-verhouding tot het werken met unit tests. Visual Studio biedt uitstekende ondersteuning, maar vaak blijkt de stap om met het opzetten van testen te starten, te groot, hoe minimaal deze ook is. De voordelen zijn nochtans, bij uitstek na verloop van tijd, heel groot. Je kan met gerust gemoed drastischer codewijzigingen doorvoeren: de set unit tests toont waar je fouten maakte. Door te starten met unit tests op je business laag kan je snel tot een applicatie komen die iets lijkt te doen wat in de goede richting ligt.

AAA patroon

Het **AAA** (Arrange, Act, Assert) patroon is een gekende manier om unit testen te schrijven voor je method:

- De **Arrange** sectie van een unit test method initialiseert objecten en kent de waarden van de gegevens toe die aan de method worden meegegeven.
- De **Act** sectie roept de te testen method op met de ingestelde parameters.
- De **Assert** sectie verifieert dat de actie van de geteste method precies beantwoordt aan het gedrag dat we verwachten.

Naming Convention

Het is een goed idee om in de naam van de unit test aan te geven wat er getest wordt. Er zijn vele "naming conventions" maar een goede is bijvoorbeeld:

- de naam van de method die getest wordt (bijvoorbeeld *Reverse*)
- gevolgd door een korte beschrijving van het test scenario (bijvoorbeeld

ShouldThrowArgumentException)

- gevolgd door de verwachting van het scenario (bijvoorbeeld *IfWordIsNull*)
- waarbij elk onderdeel gescheiden wordt door een underscore

```
public void Reverse_ShouldThrowArgumentException_IfWordIsNull()  
{  
    ...  
}
```