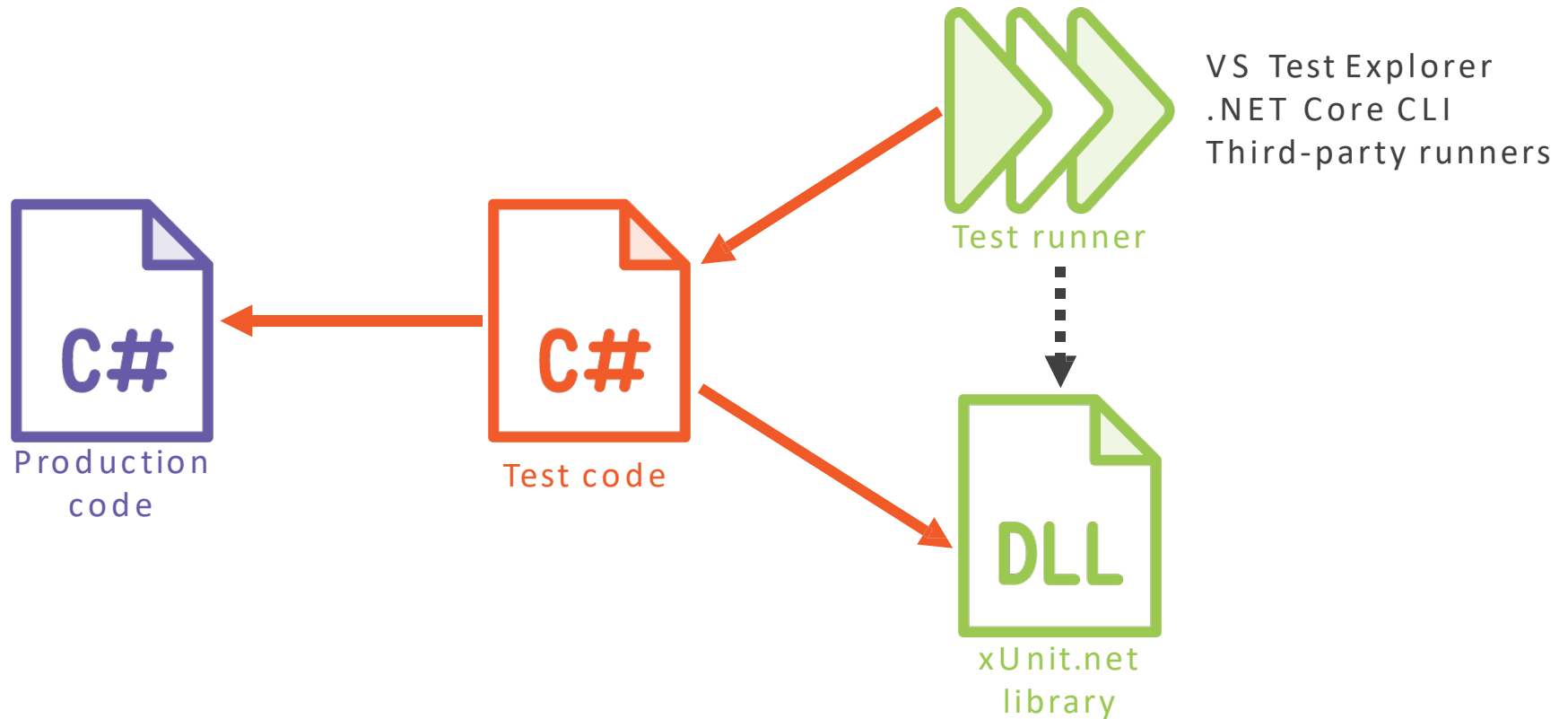


Test Driven Development

**xUnit**

# xUnit



# Ondersteunde platformen

.NET Full

.NET Core

.NET Standard

Universal  
Windows Platform  
(UWP)

Xamarin  
(iOS & Android)

<https://xunit.net/>

```
public class PlayerCharacterShould
{
```

**[Fact]**

```
public void IncreaseHealthAfterSleeping()
{
```

```
    PlayerCharacter sut = new PlayerCharacter();
```

Arrange

```
    sut.Sleep();
```

Act

```
    Assert.InRange(sut.Health, 101, 200);
```

Assert

```
}
```

**[Fact]**

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

**[Theory]**

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

```
}
```

# Asserts: falen of slagen

Hoeveel asserts per test?

De hardliners zeggen: exact 1

De pragmatici (wij ook dus) zeggen dat meerdere mogen indien ze exact hetzelfde testen

# Wat doet Assert?

- Evalueert en verifieert het resultaat van een test gebaseerd op:
  - een geretourneerd resultaat
  - de uiteindelijke toestand van een object
  - gebeurtenissen tijdens de uitvoering waargenomen
- Een assert kan slagen of falen.
- Als alle asserts slagen, slaagt de test
- Als één assert faalt, faalt de test

# Asserts

- Boolean waarden:
  - True / false
- String waarden:
  - Gelijkheid / ongelijkheid
  - Leeg
  - Start met / eindigt op
  - Bevat
  - Regular expressions
- Numerieke waarden:
  - Gelijkheid / ongelijkheid
  - In een bepaald bereik
  - Floating point precisie

# Asserts: vervolg

- Inhoud collecties:
  - Gelijkheid met andere collectie
  - Bevat wel / bevat niet
  - Bevat item dat voldoet aan predicaat
  - Alle items voldoen aan een voorwaarde
- Events:
  - Custom Events
  - Framework Events
- Object type
  - Is wel / niet exact type
  - Is wel / niet afgeleid type



```
public class PlayerCharacterShould
{
    [Fact]
    public void IncreaseHealthAfterSleeping()
    {
        PlayerCharacter sut = new PlayerCharacter();

        sut.Sleep(); Assert.InRange(sut.Health,

        101, 200);
        Assert.True(sut.IsNoob);
    }
}
```

# Asserts: voorbeelden

- `Assert.True(sut.IsNoob);`
- `Assert.StartsWith("Sarah", sut.FullName);`
- `Assert.InRange(sut.Health, 101, 200);`
- `Assert.Null(sut.Nickname);`
- `Assert.Contains("Long Bow", sut.Weapons);`
- `Assert.IsType<BossEnemy>(enemy);`
- `Assert.NotSame(enemy1, enemy2);`
- `Assert.Throws<ArgumentNullException>(() => sut.Create(null));`
- `Assert.raises<EventArgs>(...)`

# Diverse interessante links

---

[xUnit Home](#)

[xUnit Code Snippets](#)

[Vergelijking met andere frameworks](#)

[Output vangen](#)